

# DIMENSIONAL SYNTHESIS OF MECHANISMS USING DIFFERENTIAL EVOLUTION WITH AUTO-ADAPTIVE CONTROL PARAMETERS

A.ORTIZ, J.A CABRERA, F. NADAL AND A. BONILLA.

*Department of Mechanical Engineering. University of Málaga.  
C/ Doctor Ortiz Ramos s/n  
Ampliación Campus Teatinos. 29071 Málaga. Spain*

**Abstract—** This paper presents how an algorithm based on Differential Evolution (DE) with no constant control parameters solves the dimensional synthesis of four and six-bar mechanisms for path generation.

The selection of values of DE control parameters is not always an easy and obvious job. In this paper a DE with auto-adaptive control parameters is proposed which includes a new mutation operator to solve stagnation in local minima and, on top of all this, the choice of a control parameters value in a simple way.

For a set of 6 representative cases related to dimensional synthesis from bibliography, the performance of this new DE algorithm, called **Ingeniería Mecánica Málaga (IMMa)** Optimization Algorithm with Self-Adaptive Technique, IOA<sup>s-at</sup>, has been tested. An explanation of how the new mutation operator works is also included. The results obtained, compared with others synthesis techniques from literature, present significant improvements for every case studied.

The new version does not require control input parameters to be chosen by the user. Those are auto-tuned during the algorithm execution, that is, we use an algorithm with adaptive control parameters and it continues being easy-to-use, robust and fast.

**Keywords—** Synthesis of mechanisms, Differential Evolution, Genetic algorithms, Optimization methods, Self-Adaptive Differential Evolution.

## 1. INTRODUCTION.

Dimensional synthesis engages to obtain the dimensions of a chosen mechanism that must be able to satisfy a set of given requirements. It will be necessary to define the length of links, the distance among the fixed kinematic joints and the position of these with respect to the reference system.

Erdman and Sandor [1] define three types of dimensional synthesis: function generation synthesis, motion generation synthesis and path generation synthesis. In this paper, we pay attention to the last one, with and without prescribed timing. Some recent papers about this topic are Cabrera, 2002, [2], Shiakolas, 2005, [3] and Acharyya, 2009 [4]. Cabrera compared his DE algorithm with another three (Central Difference, Exact Gradient and Genetic Algorithm KK) for the synthesis of four-bar mechanisms applied to three different path and observed that DE showed faster convergence and lower error to target points than the other three algorithms. Shiakolas used DE for synthesis of six-bar mechanisms with prescribed timing when a dwell is desired at the output link. Acharyya applied three evolutionary algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and DE to three cases of path generation for 4-bar mechanisms and observed similar conclusions to Cabrera's.

Generally speaking, three different methods have been proposed to solve the path generation problem: the graphic method, the analytical method and the optimization techniques method. In this work, we focus our research on this last group, which obtains a solution that fulfills the requirements minimizing an objective function subject to a restrictions set. The importance of optimization methods is well-known due to the fact that the main industrial process and services are dependent on a limited number of variables (sometimes quite a few) and these are dependent on a big number of

parameters whose values must be selected to obtain a good behaviour of the system.

Normally, the performance of an industrial task is reproduced theoretically by means of a mathematical model. It is necessary to choose the values of the mathematical model parameters to achieve a model output as close to real physical behaviour as possible.

Evolutionary algorithms are an approach to optimization which is very extended. The most important features of this kind of algorithm are:

- I. It can start to find the global minimum independently from the initial values of the variables to be optimized.
- II. It has a high percentage of success in the optimization process.
- III. It has a fast convergence.
- IV. It needs a low number of control input parameters.

The IOA (IMMa Optimization Algorithm) uses genetic operators defined in Differential Evolution (DE). It was introduced by Storn & Price in [5]. Besides, Price, Storn and Lampinen in [6] stated that Evolution Strategies have the simplest implementation and are among the best global optimizers. Also Feoktistov said in [7]: 'without regard of its simplicity, DE was and is one of the most powerful tools for global optimization', and it presents a simple example applied to the well-known Rosenbrocks' and Rastringins' functions that demonstrate the good convergence properties and the capacity to attain the global optimum of the population-based approach versus the steepest descent methods. Besides, Golberg [8] said that convergence to the best is not an issue in business or in most walks of life; we are only concerned about finding better methods than others.

The DE algorithm is more popular every day and has been applied successfully in multiple scientific fields, because it is easy to use and understand and it has good convergence properties. The IOA algorithm presents a variant with respect to the DE scheme and introduces a new mutation operator. This operator improves the optimization procedures allowing to find the correct optimum and not to stop in a local minimum. A disadvantage of this approach has been that two new control input parameters have been introduced. So, to reduce human intervention, an auto-adapting control parameter technique is introduced in the new version of the IOA.

Here in this paper, we propose an optimization method based on Differential Evolution and without control input parameters to be selected by the user, as a tool to be employed in the optimization problems mentioned above.

This new algorithm has been used with success applied to the search of tyre model parameters [9] and it was called IIMMa Optimization Algorithm with self-adapting technique.

Now the behaviour of our algorithm has been tested for 6 cases of mechanism synthesis proposed in literature for task [2], [3], [4]. In this new optimization algorithm, the value of the input control parameters change during the algorithm execution. In bibliography, [10], [11], Eiben et al. mentioned three, very extended, different ways to realize the change:

- 1.- *Deterministic parameter control*: deterministic rules establish each parameter value.
- 2.- *Adaptive parameter control*: the direction of the search and the magnitude of the parameter change is established by means of some feedback from the search.
- 3.- *Self-adaptive parameter control*: the parameters to be adapted are encoded into the chromosome (individuals of a population) and undergo the actions of genetic operators.

In the IOA<sup>s-at</sup>, the control input parameters change during the run using the idea that the values of these parameters associated with the best individual of the population manage the optimization process to evolve the previous one. The values of the input control parameters that prevail are always those associated with the best individual of the population in every generation. Therefore, the IOA<sup>s-at</sup> may be situated in a new category halfway between the second and the third category of these. At first instance, we could think that the direction and magnitude of the DE parameter changes are obtained from some feedback of the search. However, this feedback information is in direct relationship with the actions of genetic operators. That is, we use the genetic operators to produce the control parameter changes in direction and magnitude. However, they are not encoded into the chromosome.

There are other approaches that change the control parameter values during the run. One of the most recent ones is the work of Brest et al. (2006) [12] in which the control parameters are incorporated into the chromosomes, thereby being subject to evolution. Another similar one, is the work of Quan-Ke Pan et al. (2011) [13] in which each target individual has an associated strategy list (SL), a mutation scaling factor  $F$  list (FL), and a crossover rate  $CR$  list (CRL). And during the

evolution, a trial individual is generated by using a strategy,  $F$ , and  $CR$  taken from the lists associated with the target vector. Radha et al (2009) [14], created an adaptive control parameters algorithm which, for every iteration, creates a population choosing the values of  $F$ ,  $Cr$ , and  $NP$  according to simple rules and has six fixed values of parameters ( $F_1, F_0, C_{r1}, C_{r0}, P_f, P_{Cr}$ ). All of them investigate the performance of their algorithms with a set of standard benchmark functions

In the paper presented here, the results found by applying the  $IOA^{s-at}$  to 6 cases of dimensional mechanism synthesis, are drawn. These six cases will be presented on the following pages, and they are compared with those results obtained by their respective authors.

Basically,  $IOA^{s-at}$  is a DE algorithm with auto-adaptive control parameters which uses a double trial population by means of standard DE operators mentioned later. Then, the selection operator is applied to compare the objective values among the three competing populations (the current and the two trial populations). The values of the input control parameters that survey are always those associated with the best individual of the three populations in every generation using the idea that the values of these parameters associated with the best individual of the three populations manages the optimization process to evolve the current one.

The paper is organized as follows: an overview of the problem is explained in Section 2 followed by genetic operators in DE in Section 3. In Section 2 an introduction to the problem is sketched, which motivates the requirement of a self-adapting control parameter technique in Section 4. In section 5, the objective functions for the dimensional synthesis of a four-bar and a six-bar mechanism are painted. Next, the six cases employed in this study together with the results are described in section 6 and 7. And finally, conclusions are drawn in Section 8.

## 2. PROBLEM INTRODUCCION.

In the following paragraph we are going to justify the need to create a self-adaptive algorithm based on two different concepts. On one hand, a DE control parameter ( $F$ ,  $CP$ ,  $MP$  and range) affect each other and the optimization evolution itself. And also, the new mutation operator included in the IOA influences the behaviour of the optimization evolution. On the other side, the user of an optimization method based on DE always meets the same problems. Which initial value is he going to introduce to start the optimization? And he always has the same doubt, Has the value set for DE parameters (for  $F$ ,  $CP$ ,  $MP$  and range) been chosen appropriately for this objective function?. Will it have stagnation problems before finding a global optimal solution? The agents that affect the stagnation are studied in [15].

Nowadays, nobody has found functional relationships among these four control parameters due to the fact that variations in one of them have influence on the others. As the process to reach these functional relationships to link the value of the DE parameters is not obvious, and nowadays difficult, it is the reason why we used other kinds of approaches.

In this paper, to address some drawbacks, a self-adapting control parameter technique is proposed that automatically searches the proper value of the control parameters for every generation to achieve a global optimal solution quickly. This way, choosing suitable control parameter values is a minor problematic task, i.e., it is not necessary to start the optimization with a successive trial-error process until a proper set of values for  $F$ ,  $CP, MP$  and range is found. Due to this fact, we present a technique to tune the control parameters, for each iteration, during the run.

## 3. GENETIC OPERATORS IN DIFFERENTIAL EVOLUTION.

The genetic operators used in this paper are selection, reproduction and mutation. In this work, selection and reproduction are carried out sequentially and mutation is used as an independent process.

### 3.1. Selection.

For selection, two individuals are randomly chosen from the population and they form a couple for

reproduction. The best individual and two individuals randomly selected with uniform distribution are chosen for reproduction and they make up a disturbing vector,  $V$ . The scheme, Storn and Price [5], known as differential evolution yields:

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD}) \text{ with } i \in [1, NP] \quad (1)$$

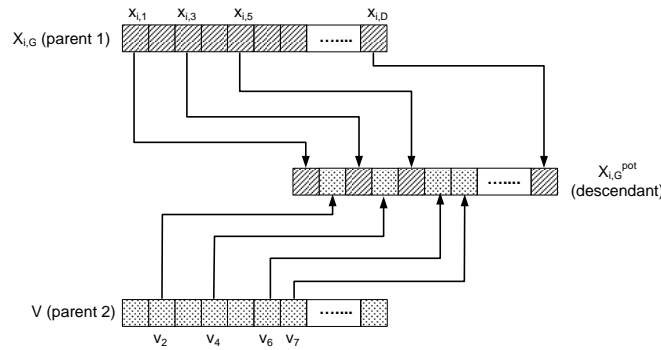
$$V = X_{best} + F \cdot (X_{r1} - X_{r2})$$

where  $X_{best}$  is the best individual of a population of  $NP$  individuals,  $X_{r1}$  and  $X_{r2}$  are two individuals randomly selected from the population, and  $F$  is a real value that controls the disturbance of the best individual, with  $F \in (0, 2)$ . The scaled difference between two randomly chosen individuals,  $F \cdot (X_{r1} - X_{r2})$ , defines direction and length of the search step. The  $F$  constant is a control parameter, which manages the trade-off between exploitation and exploration of the searching space. If  $F=0$ , the disturbing vector,  $V$ , is exactly  $X_{best}$ . However, if  $F=2$ , the disturbing vector is created adding the double distance between two randomly chosen individuals of the population to the best individual. This is the original mutation in the DE algorithm as observed in Figure 2. Note that  $V$  is an individual near the best individual, who serves as an application base point. The randomness of both search directions and base points provides the global optimum in many cases, slightly increasing convergence. However, sometimes, it is necessary to travel to far places from the best individual and for this reason we have created a new mutation operator which is explained in epigraph 3.3.

### 3.2. Reproduction.

Next, for reproduction,  $V$  is crossed with individual  $i$  of the current population to generate a potential individual  $i$  of the next population. This operator is named crossover.

In natural reproduction, parents' genes are interchanged to form the genes of their descendant or descendants.



**Figure 1: discrete multipoint crossover for reproduction.**

As is shown in Figure 1, reproduction is approached by a discrete multipoint crossover that can be used to generate  $X_{i,G}^{pot}$ : parent  $X_{i,G}$  provides its descendant with a set of genes randomly chosen over its entire chromosome and parent  $V$  provides the rest.

If the new descendant,  $X_{i,G}^{pot}$ , is better than its antecedent,  $X_{i,G}$ , the first will replace it. Otherwise,  $X_{i,G}$  is retained and  $X_{i,G}^{pot}$  is rejected. Therefore, the population neither increases nor decreases.

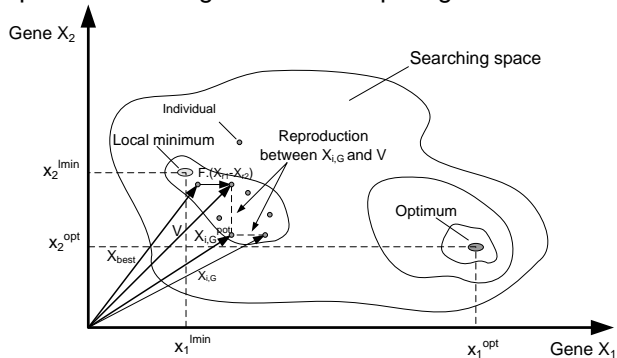
Crossover is carried out with a probability defined as  $CP \in (0, 1)$ . This means that there is a  $CP$  probability to reproduce a new 'i' descendant,  $X_{i,G}^{pot}$ , as we have explained before and a  $(1-CP)$  probability where the new 'i' descendant,  $X_{i,G}^{pot}$ , will be the same as the previous 'i' individual,  $X_{i,G}$ .

### 3.3. Mutation.

Mutation is an operator consisting of random change of a gene during reproduction. We have verified that this procedure is fundamental to obtain the optimum when the variables range values

are very different. A practical application of this has been developed for tyre model parameters identification [16] where the magnitude of every tyre model parameter is very different. The mutation procedure changes only some of these variables allowing to find the optimum and not to stop in a local minimum.

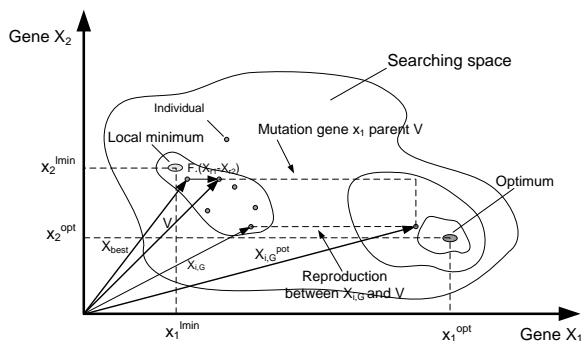
The whole procedure to obtain a new descendent without a mutation procedure is shown in Figure 2. In this case, there are two different variables (genes) and the optimum has a very different value for these two variables. We suppose that the individuals of the population are situated around a local minimum due to the evolution of the population. In Figure 2 and Figure 3 the differences between both strategies with and without mutation procedure are shown. The fundamental idea of this discussion consists of the step length adaptability along the evolutionary process. At the beginning of generations the step length is large, because individuals are far away from each other. As evolution goes on, the population converges and the step length becomes smaller and smaller.



**Figure 2: differential evolution without new mutation operator.**

In the situation mentioned above, without mutation, couple  $V$  and  $X_{i,G}$  generate descendent  $X_{i,G}^{pot}$ , but this new chromosome may not reach the global minimum due to the fact that the absolute values of the genes that compose it are very different in magnitude, and the selection plus reproduction operations are not able by themselves to make the new descendent overcome the local minimum valley.

With the mutation procedure, it is possible to solve the problem explained before. The generation of a new descendant using the mutation procedure is schemed in Figure 3.



**Figure 3: differential evolution with new mutation operator.**

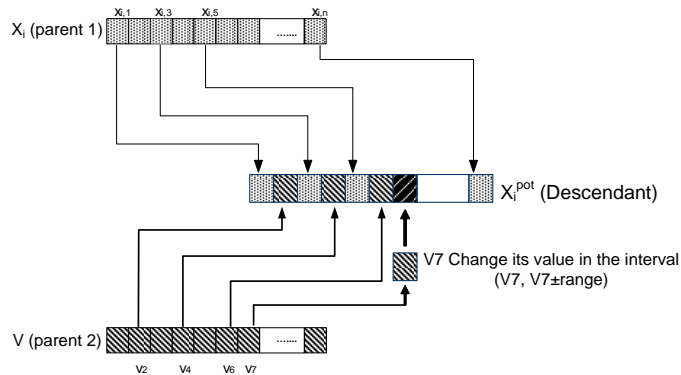
Here, the value of one or several of the genes of couple  $V$  and  $X_{i,G}$  is changed, in a range defined by the user when the reproduction is taking place. This fact yields a new descendent,  $X_{i,G}^{pot}$ , which has a different fitness from the one of descendent,  $X_{i,G}^{pot}$  studied in the previous case without mutation operator. This allows the algorithm to look for individuals with better fitness in other places of the searching space for the next generation.

In this work, mutation is defined as follows: when gene  $x_i$  mutates, the operator randomly chooses a value within the interval of real values ( $x_i, x_i \pm \text{range}$ ), which is added or subtracted from  $x_i$ , depending on the direction of the mutation as observed in Figure 4. In it, gen V7 of parent 2, changes its value in the interval ( $V7, V7+\text{range}$ ). The value of range (mutation range) depends on

the size of  $S$  (searching space) and the gen magnitude of the individuals.

Mutation is carried out with a probability defined as  $MP \in [0, 1]$ , much lower than  $CP$ , that is, there is a  $MP$  probability to change both the genes of father 1,  $X_i$ , and father 2,  $V$ . In Figure 4, only the mutation of one gene of father 2 has been shown.

Some examples where local minima were encountered without the new mutation operator are discussed in [9] (Section 3, Preliminary considerations) There IOA was run 10 times keeping  $F$ ,  $CP$  and range constant and changing the  $MP$  value from 0 to 0.9 step 0.1 to show the effect of the introduction of the new mutation operator. The  $MP$  null ( $MP=0$ ) sometimes prevents the algorithm from achieving a solution that meets the requirements.



**Figure 4: reproduction followed by mutation**

Once the genetic operators are described, the optimization algorithm will be explained.

The proposed algorithm starts with the random generation of a starting population with  $NP$  individuals. The process to generate a new population of  $NP$  individuals is the following: the best ( $X_{best}^G$ ) and two random individuals ( $X_{r1}$ ,  $X_{r2}$ ) of the actual population are chosen and the disturbing vector ( $V$ ) is formed as declared in equation (1). Then, the disturbing vector ( $V$ ) and the  $i$  individual ( $X_{i,G}$ ) are crossed and mutated with a crossover probability ( $CP$ ) and a mutation probability ( $MP$ ) respectively, yielding candidate ( $X_{i,G}^{pot}$ ) of the following population. The following step consists of evaluating candidate ( $X_{i,G}^{pot}$ ) and the  $i$  individual ( $X_{i,G}$ ) at the goal function, the best of them will go on to be the  $i$  individual in the new population ( $X_{i,G+1}$ ). This process is repeated to get the  $NP$  individuals of the new population, appearing a new best individual ( $X_{best}^{G+1}$ ) or keeping the best one ( $X_{best}^G$ ) of the previous generation. New populations are generated until one of the two following conditions is satisfied: the number of iterations reaches the maximum number of iterations or the evaluation of the best individual in the  $N$  population is lower than the minimum error.

#### 4. AUTO- ADAPTING CONTROL PARAMETER TECHNIQUE.

Choosing suitable control parameter values is a problematic task that can be avoided by successive trial-error processes, but this demands the user to carry out the multiple execution of the algorithm until a proper set of values is found. Due to this fact, we present a technique to tune the control parameters during the run here.

In this new version of the IOA, we use the same DE scheme employed in [16]. This method is the most extended one in biography [5], [17], [18], [19].

The user only has to introduce three input parameters:  $NP$  (number of individuals of the population),  $D$  and  $itermax$  (maximum number of iterations), which do not change during the run. Normally, in bibliography [10],[18],  $NP=10 \cdot D$  is suggested. And  $D$  is the number of variables to be optimized in the problem.

The task of adjusting the control parameters is realized by the IOA the following way and it is summarized in Figure 5.

The algorithm starts generating an initial random value for  $F$ ,  $CP$ ,  $MP$ , and range and they are

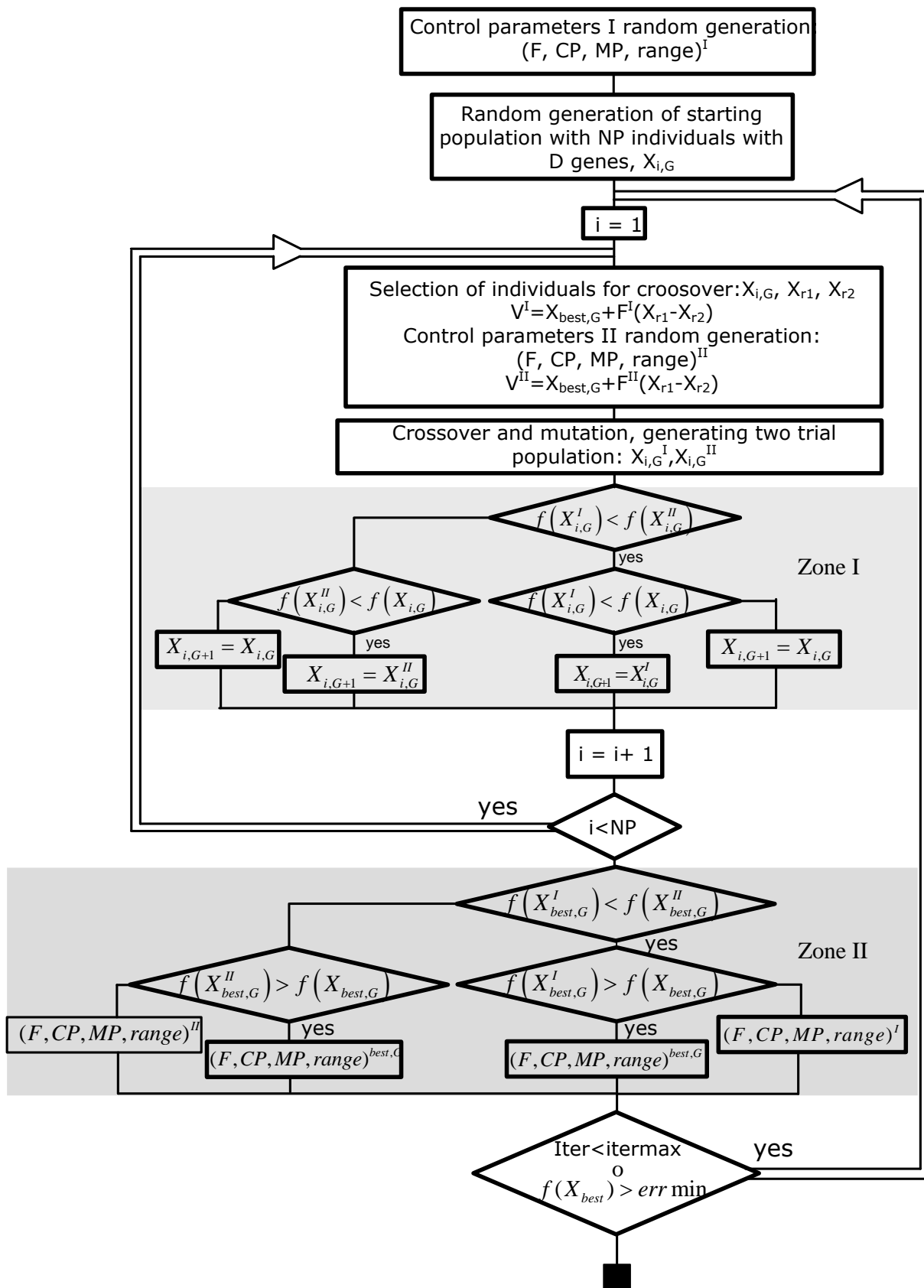


Figure 5: general Scheme of IOA<sup>s-at</sup>

used for the first generation of the first trial population. The user does not need to adjust these parameter values. Later, the algorithm continues with the random generation of a starting population with NP individuals, and every individual has D genes. The starting population ( $X_{i,G}$ ) evolves by means of the genetic operators to another two trial populations with the same number of individuals. The genetic operators are the same as presented in [8]. The first trial population, ( $X_{i,G}^I$ ), in the first iteration, is formed with the control parameters mentioned before ( $F^I$ ,  $CP^I$ ,  $MP^I$ ,  $range^I$ ) and the second, ( $X_{i,G}^{II}$ ), for all iterations, with a set of values chosen randomly as shown:

$$\begin{aligned} F^I &= 2 * rand(1) & F^{II} &= 2 * rand(1) \\ CP^I &= rand(1) & CP^{II} &= rand(1) \\ MP^I &= rand(1) & MP^{II} &= rand(1) \\ range^I &= rand(1) & range^{II} &= rand(1) \end{aligned}$$

where  $rand(1)$  is the Matlab function to generate a uniformly distributed random number between (0,1). Notice that the initial value of range is also between (0, 1), although the range is in relation to the size of the searching space, S. However, the auto-adapting control parameter technique looks for proper values for range in every case, independently of the S size.

This evolutionary process from the current population to the following generation is summarized in Figure 5. As shown in this figure (zone I), the next generation may be formed with individuals from the current population, with individuals from trial population I and with individuals from trial population II. The strategy to incorporate the individuals to the next generation is the following: the individuals are evaluated in the objective function as shown in Figure 5.

Hence, the i-individual of the three mentioned populations that has the best fitness will be the i-individual in the next generation.

This way, the population keeps the same numbers of individuals and only the best NP individuals survive for the next generation. Once the population has evolved, the process is repeated until the numbers of generations exceed the number of iterations introduced by the user.

In the following generation, to obtain the control parameters used by the genetic operators to produce trial population I from second to itermax iteration, we choose the control parameters associated with either one of the two trial populations or with the current population to contribute to the next generation with the best individual as shown in Figure 5 (zone II). In this figure,  $f(X)$  is the objective function that expresses the fitness of the X individual,  $X_{best,G}$  is the best individual of the current population,  $X_{best,G}^I$  is the best individual of the first trial population and  $X_{best,G}^{II}$  is the best individual of the second trial population, so the control parameters which yield the individual who has the best fitness, among the three mentioned before, will be the control parameters used to generate trial population I from second to itermax generation.

So the IOA finds reasonably good values for the DE control parameters, for a given objective function, although the user does not know anything about how to locate these values.

In Figure 5, the input control parameters with superscript 'I' are those associated with the best individual of trial population I, the input parameters with superscript 'best, G' are those associated with the best of the current population and with superscript 'II' are those associated with the best of trial population II.

## 5. GOAL FUNCTIONS.

Two goal functions have been used in this work. The first one evaluates the position error between target points indicated by the designer and the real position of the coupler, C, of the designed four-bar mechanism (see Figure 6):

$$f(X) = \sum_{i=1}^N \left[ (C_{Xd}^i - C_X^i)^2 + (C_{Yd}^i - C_Y^i)^2 \right] \quad (2)$$

where N is the number of target points,  $(C_{Xd}^i, C_{Yd}^i)$  and  $(C_X^i, C_Y^i)$  are the coordinates of the target and real precision points respectively. The coordinates of the real precision points are calculated using the Freudenstein equation [20] and it was indicated in [2]. Also three constraints have been

used in the optimization problem: the satisfaction of the Grashof criterion, the sequence of the crank angle and the range of the design variables. To define the complete optimization problem, the two first constraints were introduced as penalty functions, so the problem is defined as:

$$\min \left\{ \sum_{i=1}^N \left[ \left( C_{Xd}^i(X) - C_X^i \right)^2 + \left( C_{Yd}^i(X) - C_Y^i \right)^2 \right] + M_1 \cdot h_1(X) + M_2 \cdot h_2(X) \right\} \quad (3)$$

subject to:

$$x_i \in [li_i, ls_i] \quad \forall x_i \in X \quad \text{where} \quad X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0, \theta_2^1, \theta_2^2, \dots, \theta_2^N]$$

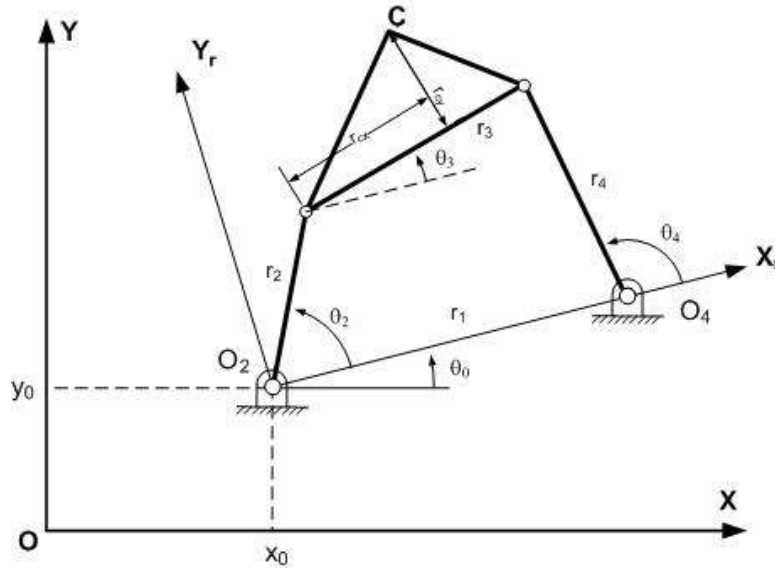


Figure 6: design variables of four-bar mechanism for the first objective function.

where  $h_1(X)$  and  $h_2(X)$  evaluate the Grashof condition and the sequence condition for the crank angle respectively, and take value 1 when those conditions are not fulfilled,  $M_1$  and  $M_2$  are the penalty factors for those functions and  $X$  represents the design variables.

The second objective function is the sum of two different functions ( $f_1$  and  $f_2$ ) and optimizes a six-bar mechanism which coupler point; C will pass through the target points while meeting the coordinated requirement between input and output angles in the dwell portion with the desired accuracy level (see Figure 7). Therefore, the objective function has a first addend ( $f_1$ ) which is formulated in the same way as equation (1):

$$f_1(X) = \sum_{i=1}^N \left[ \left( C_{Xd}^i - C_X^i \right)^2 + \left( C_{Yd}^i - C_Y^i \right)^2 \right] \quad (4)$$

where,  $N$  is the number of target points,  $(C_{Xd}^i, C_{Yd}^i)$  and  $(C_X^i, C_Y^i)$  are the coordinates of the target and real precision points of coupler C respectively. The second addend ( $f_2$ ) defines the error at the output angle for circular arc (dwell) and it can be formulated as:

$$f_2(X) = \sum_{i=1}^{N_{dp}} \left( \theta_{6d}^i - \theta_{6g}^i \right)^2 \quad (5)$$

where  $N_{dp}$  is the number of target points on dwell period, and  $\theta_{6d}$  and  $\theta_{6g}$  are the desired and generated output angles respectively. We use the Freudenstein equation [20] to obtain the coordinates of coupler C as it was indicated in [2] and the study of the output dyad to calculate the  $\theta_6$  angle.

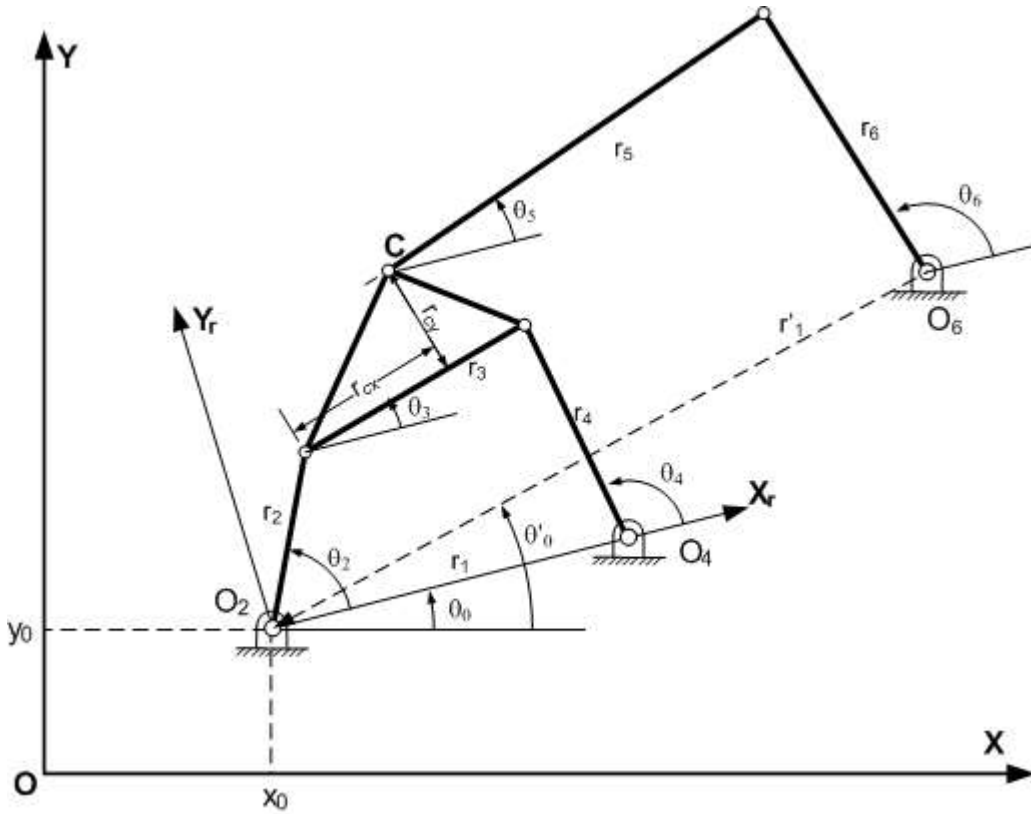


Figure 7: design variables of a six-bar mechanism for the second objective function.

Four constraints have been used in this optimization problem: the satisfaction of the Grashof criterion, the sequence of the crank angle, the range of the design variables and the non-violation of the transmission angle. The first three constraints are the same as in the first objective problem defined above. The last constraint is verified at each target point and the goal is to keep the minimum transmission angle of the mechanism larger than a desired value when the mechanism passes through those target points. The transmission angle is defined as an acute angle between the coupler and output links [21]. To define the complete optimization problem, the first two and last constraints were introduced as penalty functions, so the problem is defined as:

$$\min \left\{ \sum_{i=1}^N \left[ (C_{xd}^i(X) - C_x^i)^2 + (C_{yd}^i(X) - C_y^i)^2 \right] + \sum_{i=1}^{Nd} (\theta_{6d} - \theta_{6g}(X))^2 + \right. \\ \left. + M_1 \cdot h_1(X) + M_2 \cdot h_2(X) + M_3 \cdot h_3(X) \right\} \quad (6)$$

*subject to:*

$$x_i \in [li, ls_i] \quad \forall x_i \in X \quad \text{where } X = [r_1, r_2, r_3, r_4, r_5, r_6, r_{cx}, r_{cy}, r'_1, \theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, x_0, y_0, \theta_2^1, \theta_2^2, \dots, \theta_2^N]$$

where  $h_1(X)$ ,  $h_2(X)$  and  $h_3(X)$  evaluate the Grashof condition, the sequence condition for the crank angle and the non-violation of the transmission angle respectively,  $M_1$ ,  $M_2$  and  $M_3$  are the penalty factors for those functions and  $X$  represents the design variables.

## 6. CASES STUDIED.

The first three cases are those studied in Cabrera et al. [2]. The following two cases are the second and third case by Acharyya [4] and the last case is test case B by Shiakolas [3]. The first five cases are problems of path generation for four-bar mechanisms and the sixth case is for a six-bar

mechanism for dwell. Dwell is defined as a stationary period of the output link for non-zero input link motion.

Case 1: Path generation with six aligned coupler points and non-prescribed timing.

Case 2: Path generation with five non-aligned coupler points and with prescribed timing.

Case 3: Close loop path generation with eighteen coupler points and with prescribed timing.

Case 4: Semicircular path generation with six coupler points with prescribed timing.

Case 5: Elliptical path generation with ten coupler points and without prescribed timing.

Case 6: Two circular arcs with eighteen precision points with prescribed timing and capable of generating dwell on the six-bar mechanism output link.

For a comparison in the same frame of initial conditions as other optimization methods, the same conditions in relation to the number of goal function evaluations are used. To prove and investigate the performance and the reliability of the IOA<sup>s-at</sup>, a statistical study for every case has been made. The repetition number of the experiments was 50 runs. Also, due to the fact that IOA<sup>s-at</sup> creates two trial populations for every generation, the NP control parameters are half than those employed by other authors in their algorithms.

All the angle design variables are expressed in rad. The design variables  $[\theta^1, \theta^2, \dots, \theta^2]$  are angles measured from  $\theta_0$  positive in clockwise way.

For all the cases we calculate the structural error, defined as the difference between the prescribed position of the coupler, C and the actual generated position of the coupler, C for a given input value. However, the design error indicates the error residual incurred by a coupler point C, regarding the verification of the synthesis equations and is not discussed in this paper

### 6.1. Case 1

This case is a path generation synthesis problem, with all target points aligned and non-prescribed timing, whose solution has been frequently studied [2]. The problem is defined as follows:

- Design variables are:  $X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0, \theta^1, \theta^2, \theta^3, \theta^4, \theta^5, \theta^6]$
- Target points chosen:  $C^i_d = [(20, 20); (20, 25); (20, 30); (20, 35); (20, 40); (20, 45)]$
- Limits of the variables:  $r_1, r_2, r_3, r_4 \in [0, 60]$ ;  $r_{cx}, r_{cy}, x_0, y_0 \in [-60, 60]$ ;  $\theta_0, \theta^1, \theta^2, \theta^3, \theta^4, \theta^5, \theta^6 \in [0, 2\pi]$
- The control parameters of the IOA<sup>s-at</sup> for this case was: NP=50, D=15, itermax=1000.

The different design variables and error of one of the best mechanism for every algorithm are summarized in Table 1.

It is shown that the Grashof condition and the right sequence of input angles are fulfilled.

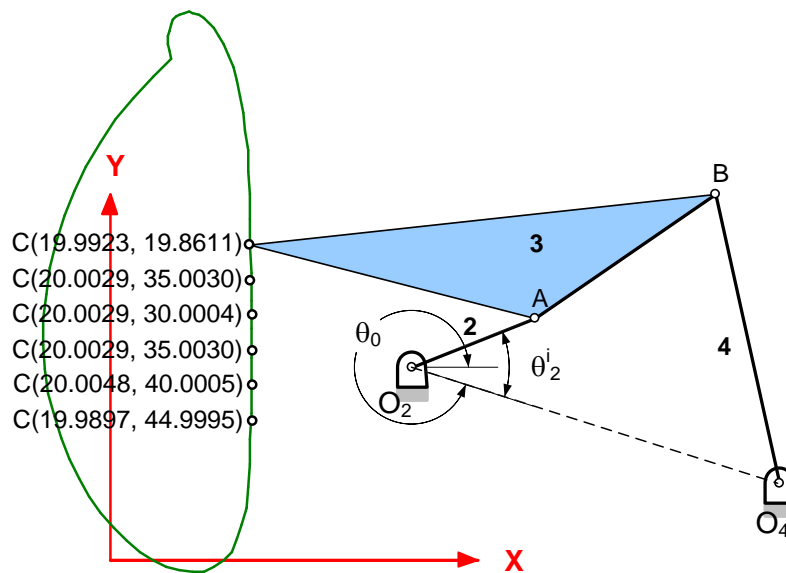


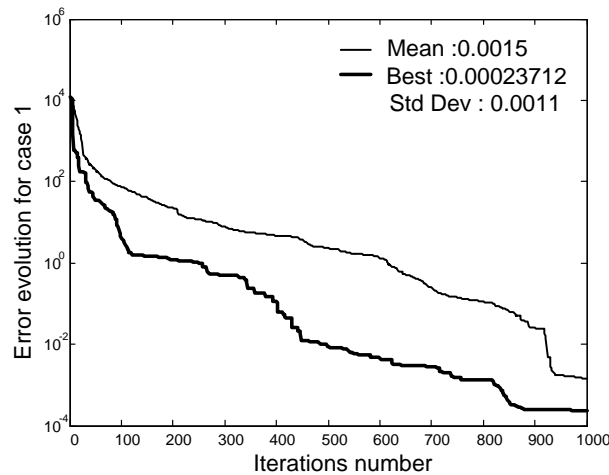
Figure 8: best mechanism for case 1.

	Cabrera et al	DE Acharyya	IOA <sup>s-at</sup>
<b>N° evaluations</b>	100000	100000	100000
$r_1$	39.46629	35.02074	54.715815
$r_2$	8.562912	6.404196	18.730993
$r_3$	19.09486	31.60722	31.223101
$r_4$	47.83886	50.59949	42.223736
$r_{cx}$	13.38556	20.80324	-27.29874
$r_{cy}$	12.21961	41.54364	31.650513
$\theta_0$	6.20163	0.000000	5.977455
$x_0$	29.7225	60.00000	43.070863
$y_0$	23.4545	18.07791	27.417061
$\theta_2^1$	6.11937	6.283185	6.424111
$\theta_2^2$	0.19304	0.264935	6.534955
$\theta_2^3$	0.44083	0.500377	0.362302
$\theta_2^4$	0.68467	0.735321	0.469063
$\theta_2^5$	0.95835	0.996529	0.577652
$\theta_2^6$	1.35533	1.333549	0.690469
<b>error</b>	0.035142	0.0178414	$2.3712 \times 10^{-4}$

**Table 1: comparative results for Case 1.**

One of the best mechanisms found in the last iteration for 50 runs is drawn in Figure 8. In this figure, the path trajectory realized by the coupler point of the four-bar mechanism is observed. And also the six positions of coupler C are shown.

The goal function evolution for the best mechanism is shown in Figure 9. Also, the mean and the Standard Deviation (Std Dev) when the IOA<sup>s-at</sup> was ran 50 times along 1000 iterations is shown in Figure 9. The standard deviation is a statistic that tells you how tightly all the final errors in 50 executions are clustered around the mean. This figure and all the figures in which the error evolution is shown have a logarithm scale for the Y-axis.



**Figure 9: evolution of the best and the mean for 50 runs for Case 1.**

## 6.2. Case 2.

The problem is a dimensional synthesis of a four-bar mechanism whose coupler point must generate a path with five non-aligned points and with prescribed timing. The four-bar mechanism has its crank-fixed point in the origin of the coordinate system and the fixed link is parallel to the X-axis, that is,  $[\theta_0, x_0, y_0] = [0, 0, 0]$ . Hence, the problem is defined as follows:

- Design variables are:  $X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}]$

- Target points chosen:  $C^i_d = [(3.000, 3.000); (2.759, 3.363); (2.372, 3.663); (1.890, 3.862); (1.355, 3.943)]$ ;  $[\theta^1_2, \theta^2_2, \theta^3_2, \theta^4_2, \theta^5_2, \theta^6_2] = [\pi/6, \pi/4, \pi/3, 5\pi/12, \pi/2]$
  - Limits of the variables:  $r_1, r_2, r_3, r_4 \in [0, 50]$ ;  $r_{cx}, r_{cy} \in [-50, 50]$ ;  $\theta_0, \theta^1_2, \theta^2_2, \theta^3_2, \theta^4_2, \theta^5_2, \theta^6_2 \in [0, 2\pi]$
  - The control parameters of the IOA<sup>s-at</sup> for this case was: NP=25, D=6, itermax=100.
- The best mechanism found for 50 executions of the IOA<sup>a-st</sup> is shown in Figure 10.

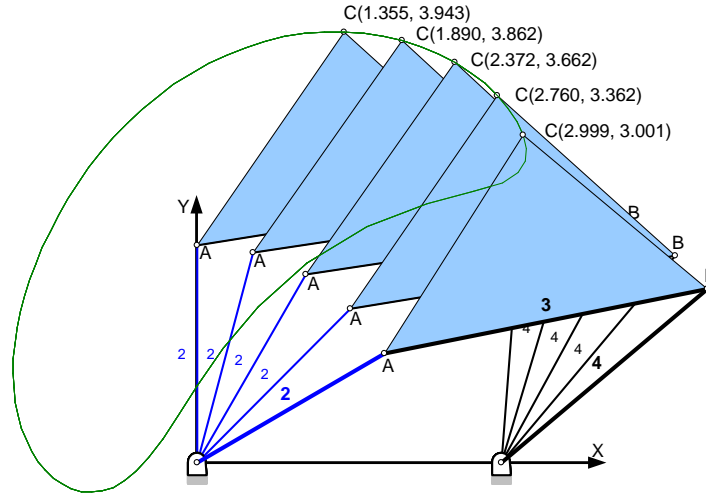


Figure 10: best Mechanism for Case 2.

The design variables that correspond to this four-bar mechanism are shown in Table 2.

	Algorithm KK	Cabrera et al	IOA <sup>s-at</sup>
<b>Nº evaluations</b>	5000	5000	5000
$r_1$	3.509643	3.0630424	2.803607
$r_2$	1.857606	1.9959624	1.99226
$r_3$	4.725835	3.305823	3.030461
$r_4$	3.518721	2.524706	2.474117
$r_{cx}$	1.959538	1.645158	1.64413
$r_{cy}$	1.558898	1.708959	1.714536
<b>error</b>	$9.429 \times 10^{-4}$	$4.075 \times 10^{-6}$	$4.2704 \times 10^{-6}$

Table 2: comparative results for Case 2

The design variables, the total number of evaluations and the final error met by Cabrera with his algorithm are also shown. Kunjur and Krishnamurty [22] also solved this case with a genetic algorithm. Their design variables and final error are added to Table 2 in the column 'Algorithm KK'.

The evolution of the error concerned when the coupler points carry out their path trajectory, measured with respect to the target points chosen above is shown in Figure 11. The evolution of the best and the mean are observed when the IOA<sup>s-at</sup> was run 50 times. The Standard deviation for these 50 runs is also shown.

The software code for this case is available in the following link:

[http://immf.uma.es/departamento/Code\\_MachineTheory.zip](http://immf.uma.es/departamento/Code_MachineTheory.zip) for researchers who might be interested to use it

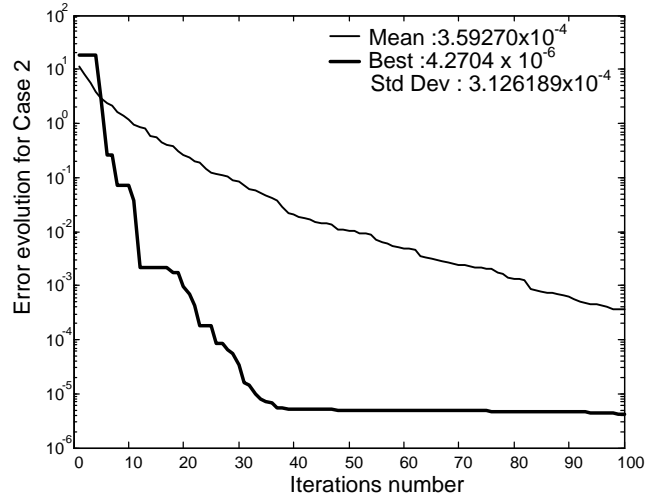


Figure 11: error evolution for Case 2, for the best and for the mean

### 6.3. Case 3.

In this dimensional synthesis problem, the coupler point of a four-bar mechanism must trace a close loop path generation with eighteen coupler points and with prescribed timing. The problem is defined as follows:

- Design variables are:  $X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0, \theta^1_2]$
- Target points chosen:  $C^i_d = [(0.5, 1.1); (0.4, 1.1); (0.3, 1.1); (0.2, 1.0); (0.1, 0.9); (0.05, 0.75); (0.02, 0.6); (0.0, 0.5); (0.0, 0.4); (0.03, 0.3); (0.1, 0.25); (0.15, 0.2); (0.2, 0.3); (0.3, 0.4); (0.4, 0.5); (0.5, 0.7); (0.6, 0.9); (0.6, 1.0)]; \theta^i_2 = [\theta^1_2, \theta^1_2 + 20 \times i]$  with  $i: 1, \dots, 17$ .
- Limits of the variables:  $r_1, r_2, r_3, r_4 \in [0, 50]; r_{cx}, r_{cy}, x_0, y_0 \in [-50, 50]; \theta_0 \in [0, 2\pi]$
- The control parameters of the IOA<sup>s-at</sup> for this case were: NP=50, D=10, itermax=50.

The best mechanism found for 50 executions of the IOA<sup>a-st</sup> is shown in Figure 12(a). The coupler path trajectory traced was drawn in Figure 12(b) where the differences from the path trajectory to the target points are observed.

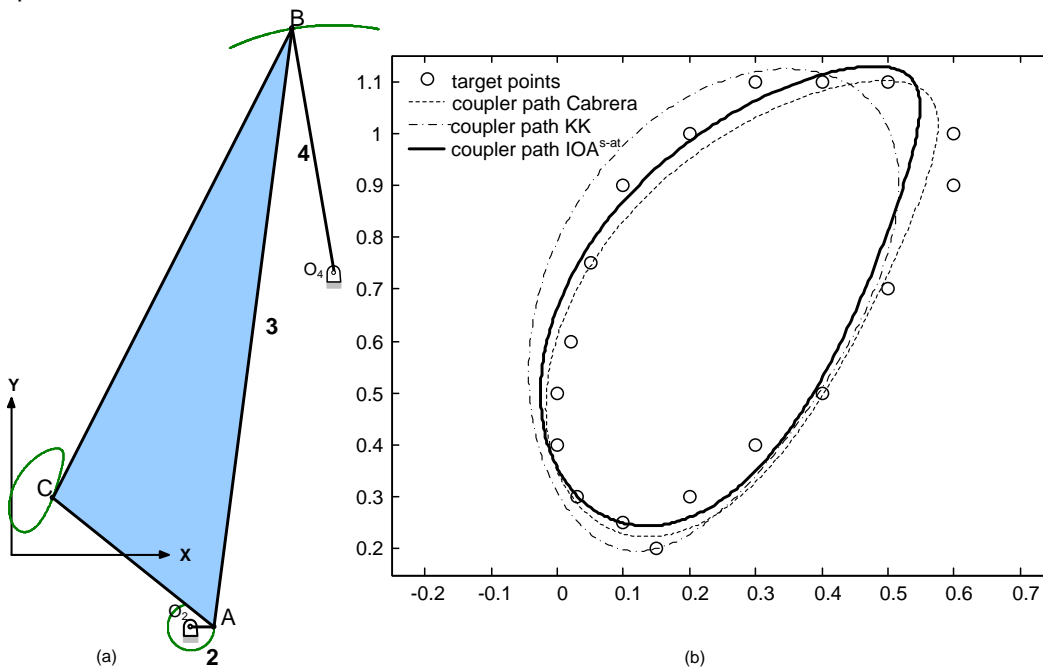


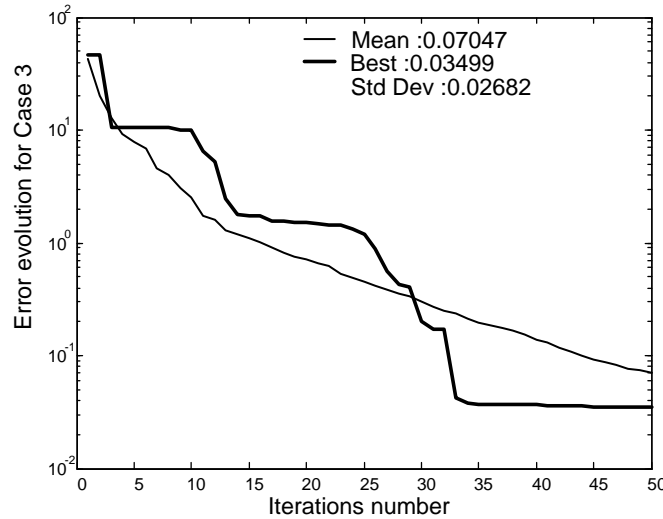
Figure 12: (a): best mechanism found for Case 3 and (b): coupler paths.

	Algorithm KK	Cabrera et al	IOA <sup>s-at</sup>
N° evaluations	5000	5000	5000
$r_1$	1.879660	3.057878	4.040435
$r_2$	0.274853	0.237803	0.245216
$r_3$	1.180253	4.828954	6.38294
$r_4$	2.138209	2.056456	2.620532
$r_{cx}$	-0.833592	0.767038	1.139106
$r_{cy}$	-0.378770	1.850828	1.866109
$\theta_0$	4.354224	1.002168	1.187751
$x_0$	1.132062	1.776808	1.891805
$y_0$	0.663433	-0.641991	-0.761339
$\theta^1_2$	2.558625	0.226186	0.000000
<b>error</b>	0.043	0.0337	0.0349

**Table 3: design variables and final error for Case 3.**

The design variables that correspond to this four-bar mechanism are shown in Table 3. The design variables, the total number of evaluations and the final error met by Cabrera with his algorithm are also shown. Kunjur and Krishnamurty [22] also solved this case with a genetic algorithm. Their design variables and final error are added to Table 2 in the column 'Algorithm KK'.

For this case, the error evolution from coupler to target points is drawn in Figure 13. The algorithm was run 50 times and the evolution of the best, mean and standard deviation were obtained.



**Figure 13: error evolution for Case 3, for the best and for the mean.**

#### 6.4. Case IV.

The fourth problem chosen here is a problem of path generation with prescribed timing. Six coupler points are chosen to find out an optimal solution that traces a semi-circular arc. This problem is solved by Acharyya [4] in his second case. Then, the problem is defined as follows:

- Design variables are:  $X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0]$
- Target points:  $C^i_d = [(0, 0); (1.9098, 5.8779); (6.9098, 9.5106); (13.09, 9.5106); (18.09, 5.8779); (20, 0)]$  and  $[\theta^1_2, \theta^2_2, \theta^3_2, \theta^4_2, \theta^5_2, \theta^6_2] = [\pi/6, \pi/3, \pi/2, 2\pi/3, 5\pi/6, \pi]$ ;
- Limits of the variables:  $r_1, r_2, r_3, r_4 \in [0, 50]$ ;  $r_{cx}, r_{cy}, x_0, y_0 \in [-50, 50]$ ;  $\theta_0 \in [0, 2\pi]$
- The control parameters of the IOA<sup>s-at</sup> for this case were: NP=50, D=9, itermax=1000.

The coupler path trajectory traced was drawn in Figure 14, where the differences from the path trajectory to the target points are observed.

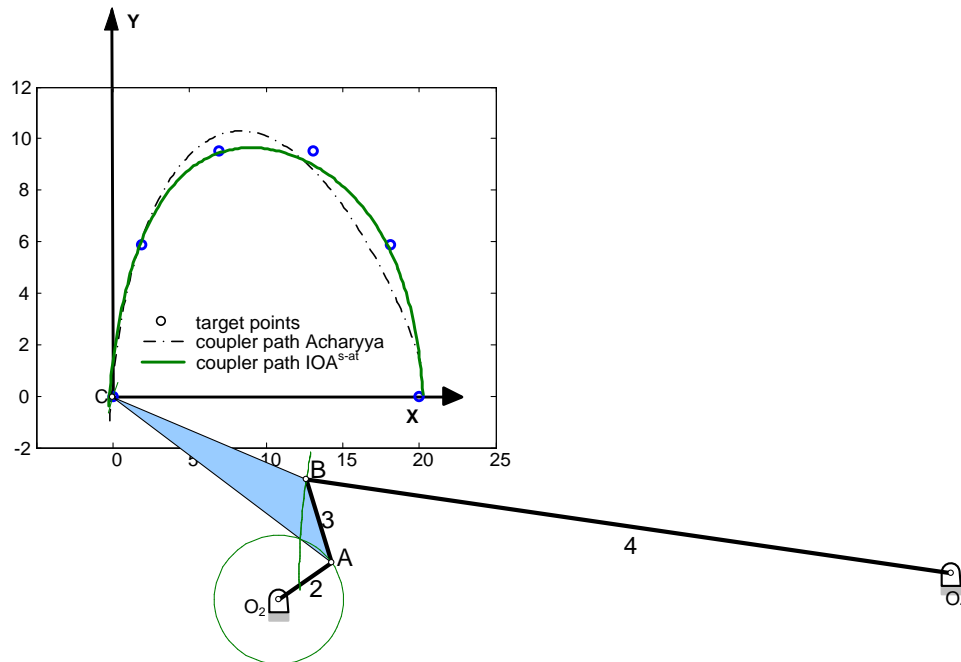


Figure 14: coupler path trajectory for every algorithm.

Also the design variables, the total number of evaluations and the final error found by Acharyya with his DE algorithm are shown in Table 4. The mechanism drawn in Figure 14 (column 3, Table 4), is the one which corresponds to the design variables found by the IOA<sup>s-at</sup> with the limits of variables mentioned in problem definition.

	Acharyya	IOA <sup>s-at</sup>
<b>N<sup>o</sup> evaluations</b>	100000	100000
$r_1$	50	49.968967
$r_2$	5	4.785659
$r_3$	5.905345	6.491026
$r_4$	50	48.393942
$r_{cx}$	18.819312	16.444782
$r_{cy}$	0	11.988091
$\theta_0$	0.463633	0.038678
$x_0$	14.373772	12.046587
$y_0$	-12.444295	-14.774897
<b>error</b>	5.52068	2.4907

Table 4: design variables for synthesis Case 4.

One of the best mechanisms found for 50 executions of the IOA<sup>a-st</sup> is shown in Table 4.

For this case, the error evolution from coupler to target points is drawn in Figure 15. The algorithm was run 50 times and the evolution of one of the best, the mean and standard deviation were obtained.

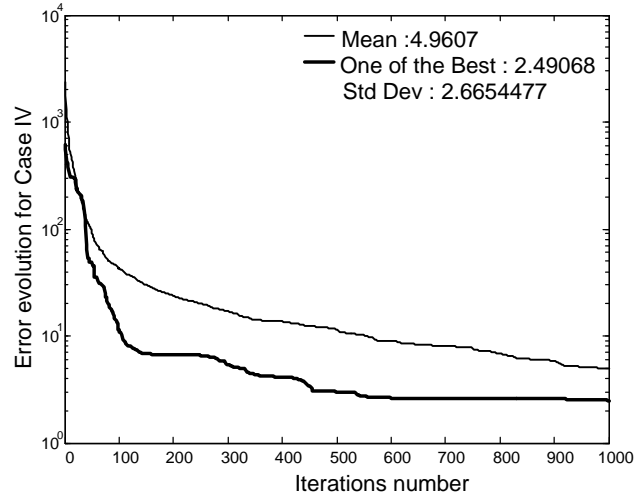


Figure 15: error evolution for Case 4.

### 6.5. Case 5.

This case is an elliptical path generation synthesis problem with ten target points and without prescribed timing. The elliptical path has a major axis of 20 units and a minor one of 16 units. The centre is kept at (10, 10) and the major axis is kept horizontal. The problem is defined as follows:

- Design variables are:  $X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0]$
- Target points:  $C^i_d = [(20, 10); (17.66, 15.142); (11.736, 17.878); (5, 16.928); (0.60307, 12.736); (0.60307, 7.2638), (5, 3.0718), (11.736, 2.1215), (17.66, 4.8577), (20, 10)]$ .
- Limits of the variables:  $r_1, r_2, r_3, r_4 \in [5, 80]$ ;  $r_{cx}, r_{cy}, x_0, y_0 \in [-80, 80]$ ;  $\theta_0, \theta^1_2, \dots, \theta^{10}_2 \in [0, 2\pi]$
- The control parameters of the IOA<sup>s-at</sup> for this case were: NP=50, D=19, itermax=1000.

The best mechanism found for 50 executions of the IOA<sup>a-st</sup> is shown in Figure 16(a). This was obtained with the limits of the variables defined before. The design variables of this four-bar mechanism are summarized in Table 5, column 3.

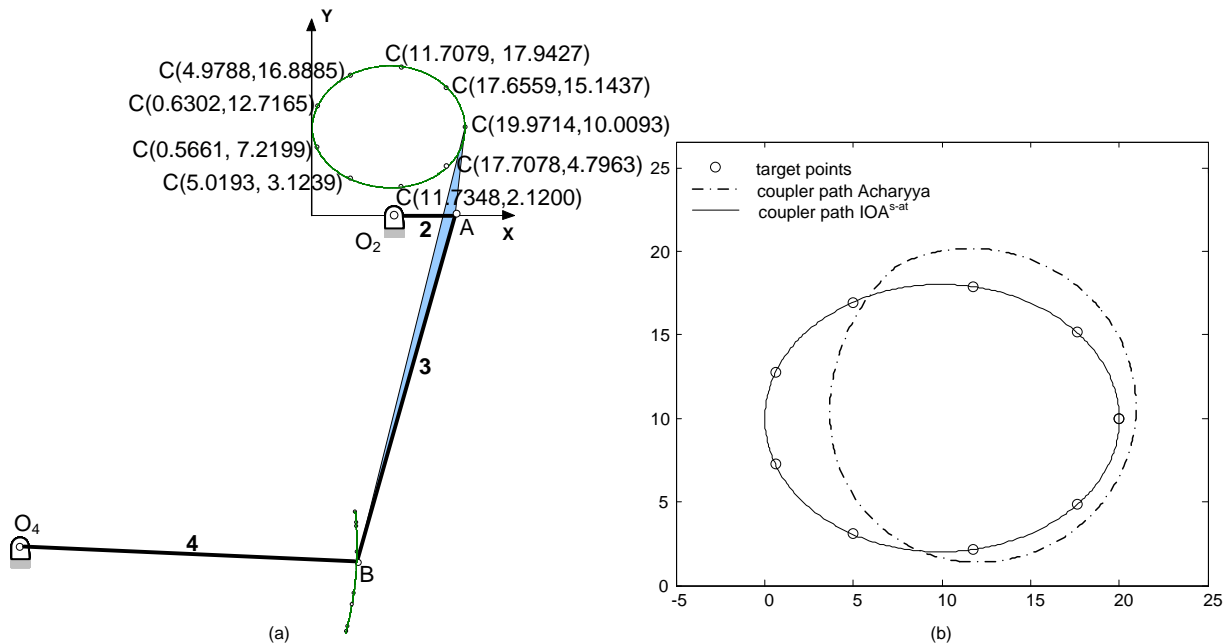


Figure 16: (a) best mechanism found with limits of variables, (b) elliptical path trajectory for every algorithm.

The coupler path trajectory traced was drawn in Figure 16(b) where differences from the path trajectory to the target points for every algorithm mentioned in Table 5 are observed.

The design variables that correspond to these four-bar mechanisms that are able to trace these elliptical paths are shown in Table 5 . Here the design variables, the total number of evaluations and the final error found by Acharyya with his DE algorithm and by the IOA<sup>s-at</sup> are shown. In this table the sequence of input angles:  $\theta_2^i$  are also shown.

	Acharyya	IOA <sup>s-at</sup>
<b>N° evaluations</b>	100000	100000
$r_1$	54.360893	65.428771
$r_2$	8.683351	8.016387
$r_3$	34.318634	47.221655
$r_4$	79.996171	44.13656
$r_{cx}$	0.000187	-11.570858
$r_{cy}$	1.46525	-1.904914
$\theta_0$	2.12965	3.86733
$x_0$	10.954397	10.635414
$y_0$	11.074534	-1.675477
$\theta_2^1$	6.283185	2.419931
$\theta_2^2$	0.616731	3.109267
$\theta_2^3$	1.310254	3.81295
$\theta_2^4$	2.19357	4.50644
$\theta_2^5$	2.91717	5.181139
$\theta_2^6$	3.490746	5.88342
$\theta_2^7$	4.132017	0.296263
$\theta_2^8$	4.922075	0.991153
$\theta_2^9$	5.695372	1.707787
$\theta_2^{10}$	6.28297	2.418865
<b>error</b>	1.952326	0.019097

Table 5: design variables for Case 5 dimensional synthesis.

The IOA<sup>s-at</sup> error evolution for the dimensional synthesis problem mentioned in Case V, is shown in Figure 17. In this figure, the mean and the best along 1000 iterations, when IOA<sup>sat</sup> was executed 50 times, is presented. Also the standard deviation for this statistical study is displayed.

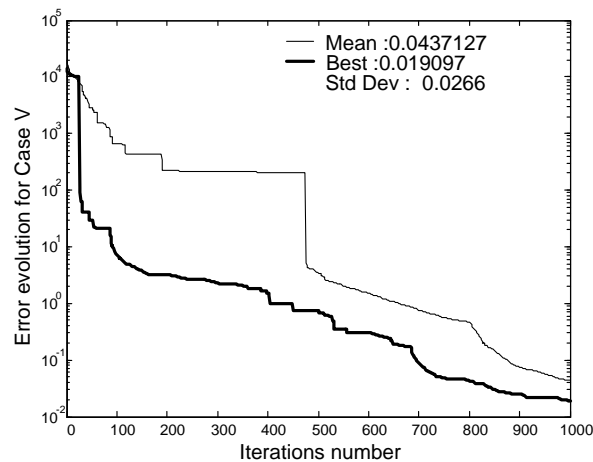


Figure 17: IOA<sup>s-at</sup> error evolution of the best and the mean for Case 5.

### 6.6. Case 6.

On this occasion, the dimensional synthesis problem consists of obtaining a six-bar mechanism whose coupler point was capable of tracing two circular arcs, with eighteen precision points, with prescribed timing and capable of generating a dwell on the six-bar mechanism. This problem is solved by Shiakolas et al. [3] in his test case B. Then, the problem is defined as follows:

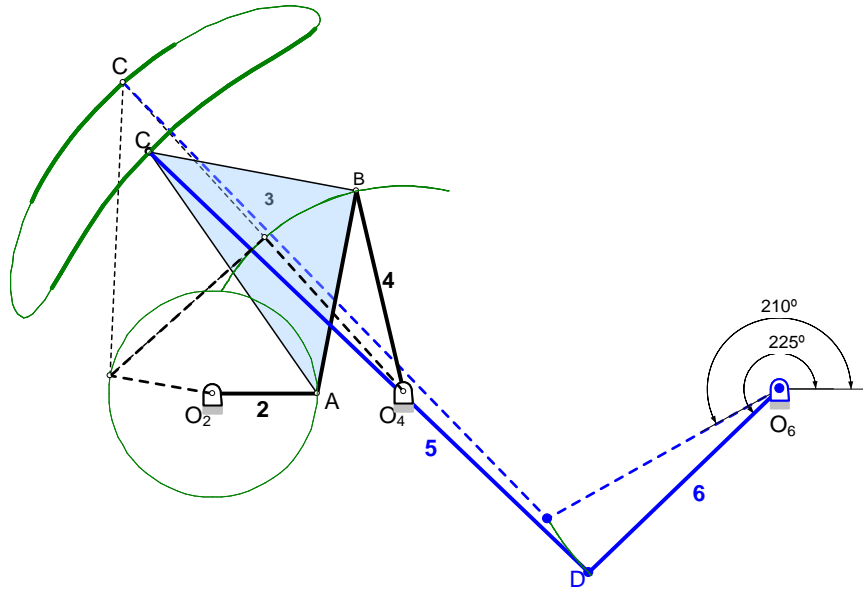
- Design variables are:  $X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0, r_5, r_6, r_1', \theta_0', \theta_2]$
- Target points:  $C^i_d = [(-0.5424, 2.3708); (0.2202, 2.9871); (0.9761, 3.4633); (1.0618, 3.638); (0.8835, 3.7226); (0.5629, 3.7156); (0.1744, 3.6128); (-0.2338, 3.4206); (-0.6315, 3.1536); (-1, 2.8284); (-1.3251, 2.46); (-1.5922, 2.0622); (-1.7844, 1.6539); (-1.8872, 1.2654); (-1.8942, 0.9448); (-1.8096, 0.7665); (-1.6349, 0.8522); (-1.1587, 1.6081)]$  and  $[\theta^1_2, \theta^2_2, \theta^3_2, \theta^4_2, \theta^5_2, \theta^6_2, \theta^7_2, \theta^8_2, \theta^9_2, \theta^{10}_2, \theta^{11}_2, \theta^{12}_2, \theta^{13}_2, \theta^{14}_2, \theta^{15}_2, \theta^{16}_2, \theta^{17}_2, \theta^{18}_2] = [0, \pi/12, 2\pi/9, \pi/3, 4\pi/9, 5\pi/9, 2\pi/3, 7\pi/9, 8\pi/9, \pi, 10\pi/9, 11\pi/9, 4\pi/3, 13\pi/9, 14\pi/9, 5\pi/3, 16\pi/9, 23\pi/12]$ ;
- Limits of the variables:  $r_1, r_2, r_3, r_4, r_5, r_6, r_1' \in [0, 5]$ ;  $r_{cx}, r_{cy}, x_0, y_0 \in [-5, 5]$ ;  $\theta_0, \theta_0', \theta_2 \in [0, 2\pi]$
- The control parameters of the IOA<sup>s-at</sup>, for this case was: NP=50, D=14, itermax=1000.

The design variables of the six-bar mechanism found using IOA<sup>s-at</sup> are summarized in Table 6, column 3. These design variables are the best found when the IOA<sup>s-at</sup> was executed 50 times. Also, the number of evaluations of the goal function and the final error is shown.

	Shiakolas	IOA <sup>s-at</sup>
<b>Nº evaluations</b>	93405	93405
<b>r<sub>1</sub></b>	0.9826	1.810744
<b>r<sub>2</sub></b>	2.0177	0.991753
<b>r<sub>3</sub></b>	2.0009	1.987899
<b>r<sub>4</sub></b>	1.8065	1.982751
<b>r<sub>cx</sub></b>	2.0336	1.989389
<b>r<sub>cy</sub></b>	2.02675	2.012227
<b>θ<sub>0</sub></b>	0.6636*π/180	0.001105
<b>x<sub>0</sub></b>	0.0415	0.013322
<b>y<sub>0</sub></b>	-0.0377	0.002008
<b>r<sub>5</sub></b>	5.7769	5.825138
<b>r<sub>6</sub></b>	2.5296	2.540682
<b>r'<sub>1</sub></b>	5.2817	5.394427
<b>θ'<sub>0</sub></b>	0.9758*π/180	0.010244
<b>θ<sub>2</sub></b>	0.2090*π/180	6.280025
<b>error</b>	0.0074	2.130796×10 <sup>-4</sup>

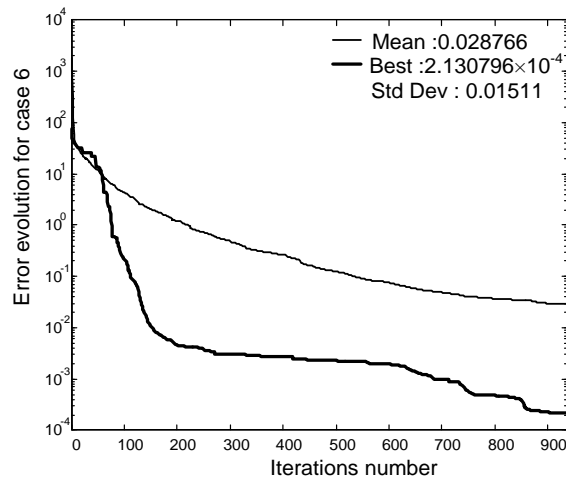
**Table 6: design variables for Case 6 dimensional synthesis.**

The best mechanism found in the last iteration for 50 runs is drawn in Figure 18. In this figure, the path trajectory realized by the coupler point of the six-bar mechanism is observed. The part of the C trajectory that corresponds to the angular position of link 6 where dwell behaviour is produced is also observed. These parts are highlighted in the figure.



**Figure 18: coupler trajectory and dwells of the six-bar found using IOA<sup>s-at</sup>**

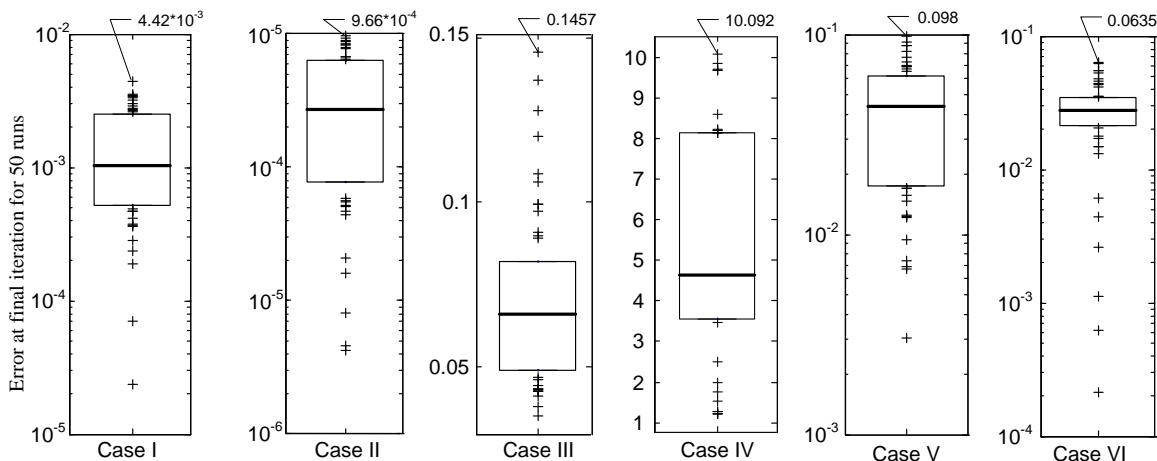
The goal function evolution for the best mechanism is shown in Figure 9. Also, the mean and the Standard Deviation (Std Dev) when the IOA<sup>s-at</sup> was run 50 times along 1000 iterations is shown in Figure 19.



**Figure 19: IOA<sup>s-at</sup> error evolution of the best and the mean for Case 6.**

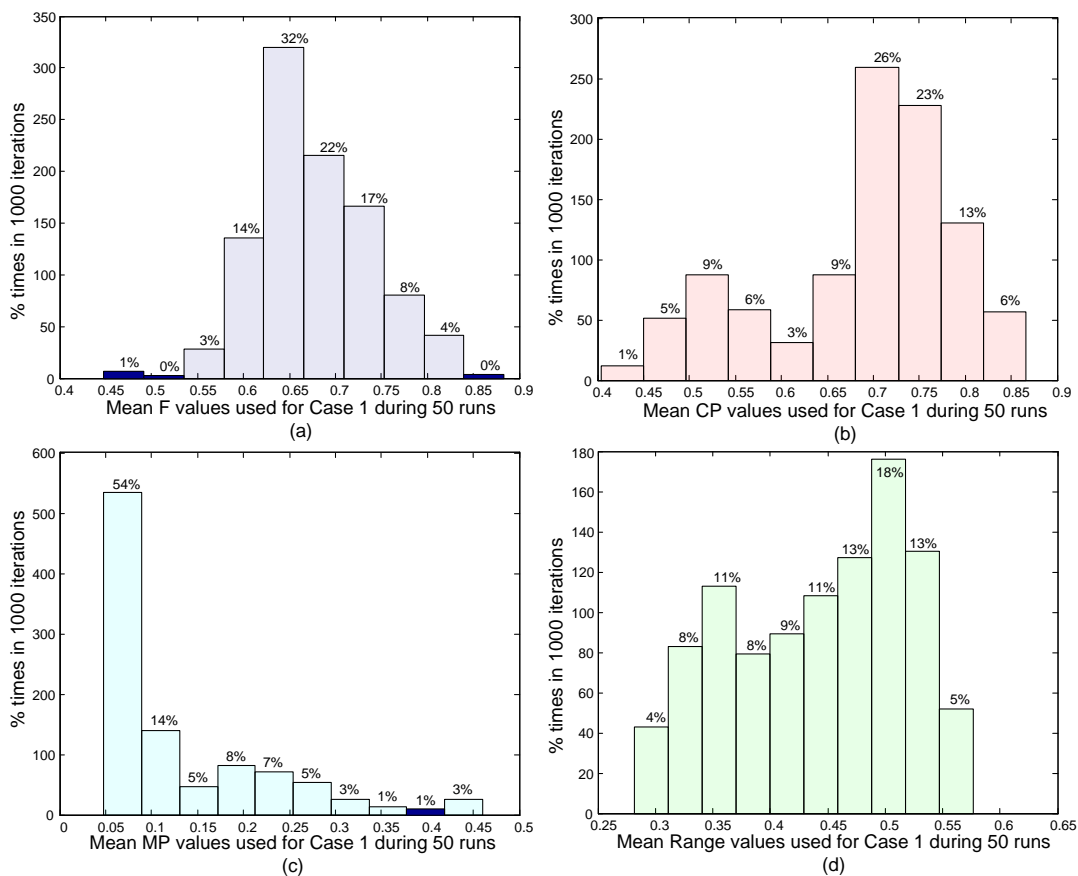
## 7. STATISTICAL ANALYSIS OF RESULTS

Finally, to investigate the performance of the algorithms, we made a statistical study. We carried out 50 different runs of the algorithm and, each time the algorithm was run, the error value to target points was recorded. Figure 20 shows a box plot for every case studied. The superior line of the box is the upper quartile error value, the inferior line of the box is the lower quartile error value and the line inside the box is the median error and the rest of the data are represented outside these boxes. Then, we can observe the best result obtained for 50 executions of the algorithm and the dispersion of the final error. In this figure, except for cases III and IV, the box plots are drawn with a logarithm scale for Y-axis. The worst final error of the 50 run for every case has been pointed out in this figure too.



**Figure 20: statistical study for the six cases studied.**

In Figure 21, the values of the four control parameters (F, CP, MP and range) during the carried out optimisation process for Case 1 have been represented. In the same way the control parameter values for the other five cases can be recorded and plotted. The percentage of F values most repeated are included between [0.6 , 0.8] as shown in Figure 21(a). Normally, the MP is much lower than CP as observed in Figure 21 (b) and (c). The most frequent values for the control parameters for Case 1 are: F=0.65, CP=0.7, MP=0.05 and Range: 0.5.



**Figure 21: values and times rate that this value occurs of F, CP, MP, Range for Case 1.**

## 8. CONCLUSIONS.

We present an optimization algorithm, called IOA, based on DE, to which a new mutation operator has been added and a double trial population, with the purpose of changing the DE parameters during the run, is used. This way, the final result is an optimization algorithm called IOA with a self-adapting control technique. Here, in this paper, the behaviour and performance of the IOA<sup>s-at</sup> have been tested for six dimensional synthesis cases.

One relevant IOA<sup>s-at</sup> characteristic is that the problematic task of choosing suitable control parameter values to start the optimization is avoided. Hence, the user does not need to carry out the multiple execution of the algorithm until a proper set of values is found. The IOA<sup>s-at</sup> presented here, tunes the control parameters during the run.

Other elements to be noticed are the simplicity of the algorithm implementation and the start of the optimization process without input control parameters and independently from the initial value of the variables to be optimized.

In full view of the results, the IOA<sup>s-at</sup> presents good behaviour in getting the optimum for whatever kind of case studied, whatever the number of precession points and the trajectory (straight line or curve) and with and without prescribed timing. Comparison with the other methods employed in literature has also demonstrated the IOA<sup>s-at</sup> performance quality in getting the optimum.

Therefore, the features of the algorithm presented here are:

It can start to find the global minimum, independently from the initial values of the variables to be optimized. Here, these values were taken between (0, 1).

The success rate of reaching a global optimum is high for all the cases studied.

It has a fast convergence as shown in the error evolution along the iterations.

It does not need control input parameters to be decided by the user.

It only needs three input parameters: NP: number of individuals of a population, D: number of variables to be optimized and itermax: maximum number of iterations.

## 8. REFERENCES.

- [1] G.N. Sandor, A.G. Erdman, *Advanced Mechanism Design. Analysis and Synthesis*, Prentice-Hall, New Jersey, 1984.
- [2] J.A. Cabrera, A. Simon, M. Prado, *Optimal synthesis of mechanisms with genetic algorithms*, Mechanism and Machine Theory 37 (2002) 1165–1175.
- [3] P.S. Shiakolas, D. Koladiya, J. Kebrle, On the optimum synthesis of six-bar linkages using differential evolution and the geometric centroid of precision positions, Mechanism and Machine Theory 40 (2005) 319–335.
- [4] S.K. Acharyya, M Mandal, *Performance of EAs for four-bar linkage synthesis*, Mechanism and Machine Theory, 44 (2009), 1784-1794.
- [5] R. Storn and K. Price, Differential Evolution. *A Simple and Efficient Heuristic Scheme for Global Optimization over Continuous Spaces*, Journal of Global Optimization 11 (1997), pp. 341-359.
- [6] K. Price, R. Storn and J. Lampinen, *Differential Evolution. A practical approach to global optimization*, Chapter 1, 'The motivation for Differential Evolution', ISBN: 3540209506, Springer, 2005.
- [7] V. Feoktistov, *Differential Evolution. In search of solutions*, Chapter 1, 'Differential Evolution', ISBN: 0387368957, Springer, 2006.
- [8] D.E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Chapter 1, 'A gentle introduction to genetic algorithms', ISBN: 0201157675, Addison-Wesley, 1989,
- [9] Ortiz, A., Cabrera, J. A., Guerra, A. and Simon, A. *The IMMα optimisation algorithm without control input parameters*, Vehicle System Dynamics, vol:47, n° 2, pp.243 — 264. (2009).
- [10] A. E. Eiben, R. Hinterding, and Z. Michalewicz, *Parameter control in evolutionary algorithms*, IEEE Trans. Evol. Comput., vol. 3, no. 2, pp.124–141, Jul. 1999.
- [11] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing. Berlin, Germany: Springer-Verlag, 2003.

- [12] Brest, J., Greiner, S., Boskovic, B, Mernik, M., and Zumer, V. *Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems*. IEEE Transactions On Evolutionary Computation, Vol. 10, No. 6. pp. 646-657, December 2006.
- [13] Quan-Ke Pan., P.N. Suganthan., Ling Wang., Liang Gao., R. Mallipeddi. *A differential evolution algorithm with self-adapting strategy and control parameters*, Computers & Operations Research 38 (2011), 394-408.
- [14] Radha Thangaraj., Millie Pant., and Ajith Abraham., *A simple adaptive Differential Evolution algorithm*. World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009. doi:10.1109/NABIC.2009.5393350
- [15] J. Lampinen, I. Zelinda, *On stagnation of the differential evolution algorithm*, Proceedings of the 6<sup>th</sup> international Mendel conference on soft computing (2000), pp. 76-83.
- [16] J.A.Cabrera, A. Ortiz, E. Carabias and A. Simon, *An alternative method to determine the Magic tyre model parameters using genetic algorithms*. Vehicle System Dynamics, vol. 41, n<sup>o</sup> 2, (2004), pp. 109–127.
- [17] J. Liu and J. Lampinen, *A fuzzy adaptive differential evolution algorithm*, Soft Computing – A fusion of foundations, Methodologies and applications, vol. 9, n<sup>o</sup> 6, (2005), pp. 448-462.
- [18] J. Sun, Q. Zhang and E. Tsang, *DE/EDA: A new evolutionary algorithm for global optimization*, Info. Sci., vol. 169, (2004), pp. 249–262.
- [19] R. Gämperle, S.D. Müller and P. Koumoutsakos, *A parameter study for differential evolution*, WSEAS NNA-FSFS-EC. Interlaken, Switzerland, WSEAS, (2002), Feb. 11-15.
- [20] F. Freudenstein, *An Analytical Approach to the Design of Four-Link Mechanisms*, Transactions of the ASME 76 (1954) 483-492.
- [21] G. Erdman, G.N. Sandor, *Mechanism Design: Analysis and Synthesis*, Vol. I, second ed., Prentice-Hall Inc., Englewood Cliffs, NJ, 1991.
- [22] A. Kunjur, S. Krishnamurty, *Genetic algorithms in mechanical synthesis*, Journal of Applied Mechanisms and Robotics 4 (2) (1997) 18–24.

**Figure captions page:**

<b>Figure 1: discrete multipoint crossover for reproduction.</b>	<b>Page 4</b>
<b>Figure 2: differential evolution without mutation operator.</b>	<b>Page 5</b>
<b>Figure 3: differential evolution with mutation operator.</b>	<b>Page 5</b>
<b>Figure 4: reproduction followed by mutation</b>	<b>Page 6</b>
<b>Figure 5: general Scheme of IOA<sup>s-at</sup></b>	<b>Page 7</b>
<b>Figure 6: design variables of four-bar mechanism for the first objective function.</b>	<b>Page 9</b>
<b>Figure 7: design variables of a six-bar mechanism for the second objective function.</b>	<b>Page 10</b>
<b>Figure 8: best mechanism for case 1.</b>	<b>Page 11</b>
<b>Figure 9: evolution of the best and the mean for 50 runs for Case 1.</b>	<b>Page 12</b>
<b>Figure 10: best Mechanism for Case 2.</b>	<b>Page 13</b>
<b>Figure 11: error evolution for Case 2, for the best and for the mean</b>	<b>Page 14</b>
<b>Figure 12: (a): best mechanism found for Case 3 and (b): coupler paths.</b>	<b>Page 14</b>
<b>Figure 13: error evolution for Case 3, for the best and for the mean.</b>	<b>Page 15</b>
<b>Figure 14: coupler path trajectory for every algorithm.</b>	<b>Page 16</b>
<b>Figure 15: error evolution for Case 4.</b>	<b>Page 17</b>
<b>Figure 16: (a) best mechanism found with limits of variables, (b) elliptical path trajectory for every algorithm.</b>	<b>Page 17</b>
<b>Figure 17: IOA<sup>s-at</sup> error evolution of the best and the mean for Case 5.</b>	<b>Page 18</b>
<b>Figure 18: coupler trajectory and dwells of the six-bar found using IOA<sup>s-at</sup></b>	<b>Page 20</b>
<b>Figure 19: IOA<sup>s-at</sup> error evolution of the best and the mean for Case 6.</b>	<b>Page 20</b>
<b>Figure 20: statistical study for the six cases studied.</b>	<b>Page 21</b>
<b>Figure 22: values and times rate that this value occurs of F, CP, MP, Range for Case 1.</b>	<b>Page 21</b>

	<b>Cabrera et al</b>	<b>DE Acharyya</b>	<b>IOA<sup>s-at</sup></b>
<b>N° evaluations</b>	100000	100000	100000
$r_1$	39.46629	35.02074	54.715815
$r_2$	8.562912	6.404196	18.730993
$r_3$	19.09486	31.60722	31.223101
$r_4$	47.83886	50.59949	42.223736
$r_{CX}$	13.38556	20.80324	-27.29874
$r_{CY}$	12.21961	41.54364	31.650513
$\theta_0$	6.20163	0.000000	5.977455
$x_0$	29.7225	60.00000	43.070863
$y_0$	23.4545	18.07791	27.417061
$\theta_2^1$	6.11937	6.283185	6.424111
$\theta_2^2$	0.19304	0.264935	6.534955
$\theta_2^3$	0.44083	0.500377	0.362302
$\theta_2^4$	0.68467	0.735321	0.469063
$\theta_2^5$	0.95835	0.996529	0.577652
$\theta_2^6$	1.35533	1.333549	0.690469
<b>error</b>	0.035142	0.0178414	$2.3712 \times 10^{-4}$

Table 1: comparative results for Case 1.

	<b>Algorithm KK</b>	<b>Cabrera et al</b>	<b>IOA<sup>s-at</sup></b>
<b>N° evaluations</b>	5000	5000	5000
$r_1$	3.509643	3.0630424	2.803607
$r_2$	1.857606	1.9959624	1.99226
$r_3$	4.725835	3.305823	3.030461
$r_4$	3.518721	2.524706	2.474117
$r_{CX}$	1.959538	1.645158	1.64413
$r_{CY}$	1.558898	1.708959	1.714536
<b>error</b>	$9.429 \times 10^{-4}$	$4.075 \times 10^{-6}$	$4.2704 \times 10^{-6}$

Table 2: comparative results for Case 2 .

	<b>Algorithm KK</b>	<b>Cabrera et al</b>	<b>IOA<sup>s-at</sup></b>
<b>N° evaluations</b>	5000	5000	5000
$r_1$	1.879660	3.057878	4.040435
$r_2$	0.274853	0.237803	0.245216
$r_3$	1.180253	4.828954	6.38294
$r_4$	2.138209	2.056456	2.620532
$r_{CX}$	-0.833592	0.767038	1.139106
$r_{CY}$	-0.378770	1.850828	1.866109
$\theta_0$	4.354224	1.002168	1.187751
$x_0$	1.132062	1.776808	1.891805
$y_0$	0.663433	-0.641991	-0.761339
$\theta_2^1$	2.558625	0.226186	0.000000
<b>error</b>	0.043	0.0337	0.0349

Table 3: design variables and final error for Case 3.

	Acharyya	IOA <sup>s-at</sup>
<b>N° evaluations</b>	100000	100000
$r_1$	50	49.968967
$r_2$	5	4.785659
$r_3$	5.905345	6.491026
$r_4$	50	48.393942
$r_{cx}$	18.819312	16.444782
$r_{cy}$	0	11.988091
$\theta_0$	0.463633	0.038678
$x_0$	14.373772	12.046587
$y_0$	-12.444295	-14.774897
<b>error</b>	5.52068	2.4907

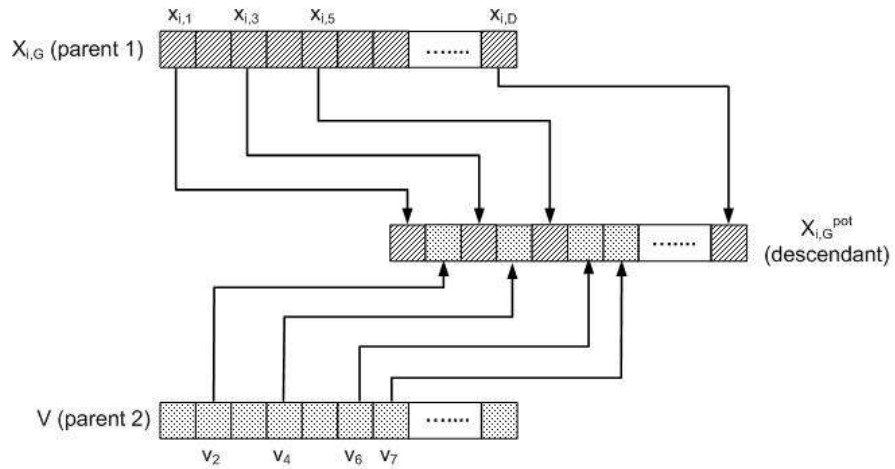
Table 4: design variables for synthesis Case 4.

	Acharyya	IOA <sup>s-at</sup>
<b>N° evaluations</b>	100000	100000
$r_1$	54.360893	65.428771
$r_2$	8.683351	8.016387
$r_3$	34.318634	47.221655
$r_4$	79.996171	44.13656
$r_{cx}$	0.000187	-11.570858
$r_{cy}$	1.46525	-1.904914
$\theta_0$	2.12965	3.86733
$x_0$	10.954397	10.635414
$y_0$	11.074534	-1.675477
$\theta^1_2$	6.283185	2.419931
$\theta^2_2$	0.616731	3.109267
$\theta^3_2$	1.310254	3.81295
$\theta^4_2$	2.19357	4.50644
$\theta^5_2$	2.91717	5.181139
$\theta^6_2$	3.490746	5.88342
$\theta^7_2$	4.132017	0.296263
$\theta^8_2$	4.922075	0.991153
$\theta^9_2$	5.695372	1.707787
$\theta^{10}_2$	6.28297	2.418865
<b>error</b>	1.952326	0.019097

Table 5: design variables for Case 5 dimensional synthesis.

	<b>Shiakolas</b>	<b>IOA<sup>s-at</sup></b>
<b>N<sup>o</sup> evaluations</b>	93405	93405
<b>r<sub>1</sub></b>	0.9826	1.810744
<b>r<sub>2</sub></b>	2.0177	0.991753
<b>r<sub>3</sub></b>	2.0009	1.987899
<b>r<sub>4</sub></b>	1.8065	1.982751
<b>r<sub>CX</sub></b>	2.0336	1.989389
<b>r<sub>CY</sub></b>	2.02675	2.012227
<b>θ<sub>0</sub></b>	0.6636*π/180	0.001105
<b>x<sub>0</sub></b>	0.0415	0.013322
<b>y<sub>0</sub></b>	-0.0377	0.002008
<b>r<sub>5</sub></b>	5.7769	5.825138
<b>r<sub>6</sub></b>	2.5296	2.540682
<b>r'<sub>1</sub></b>	5.2817	5.394427
<b>θ'<sub>0</sub></b>	0.9758*π/180	0.010244
<b>θ<sub>2</sub></b>	0.2090*π/180	6.280025
<b>error</b>	0.0074	2.130796×10 <sup>-4</sup>

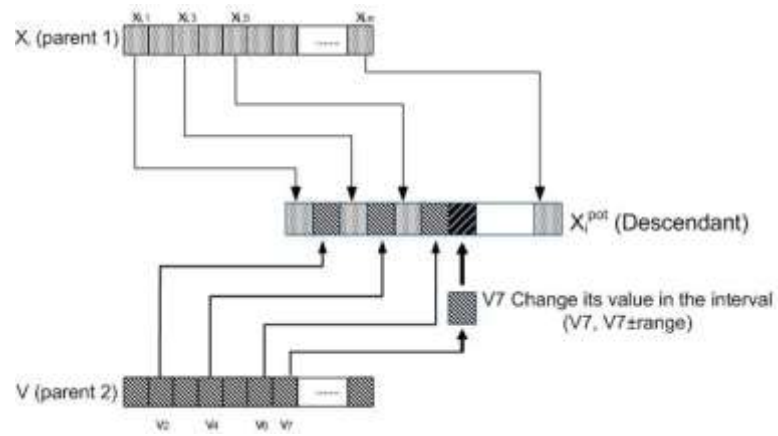
**Table 6: design variables for Case 6 dimensional synthesis.**



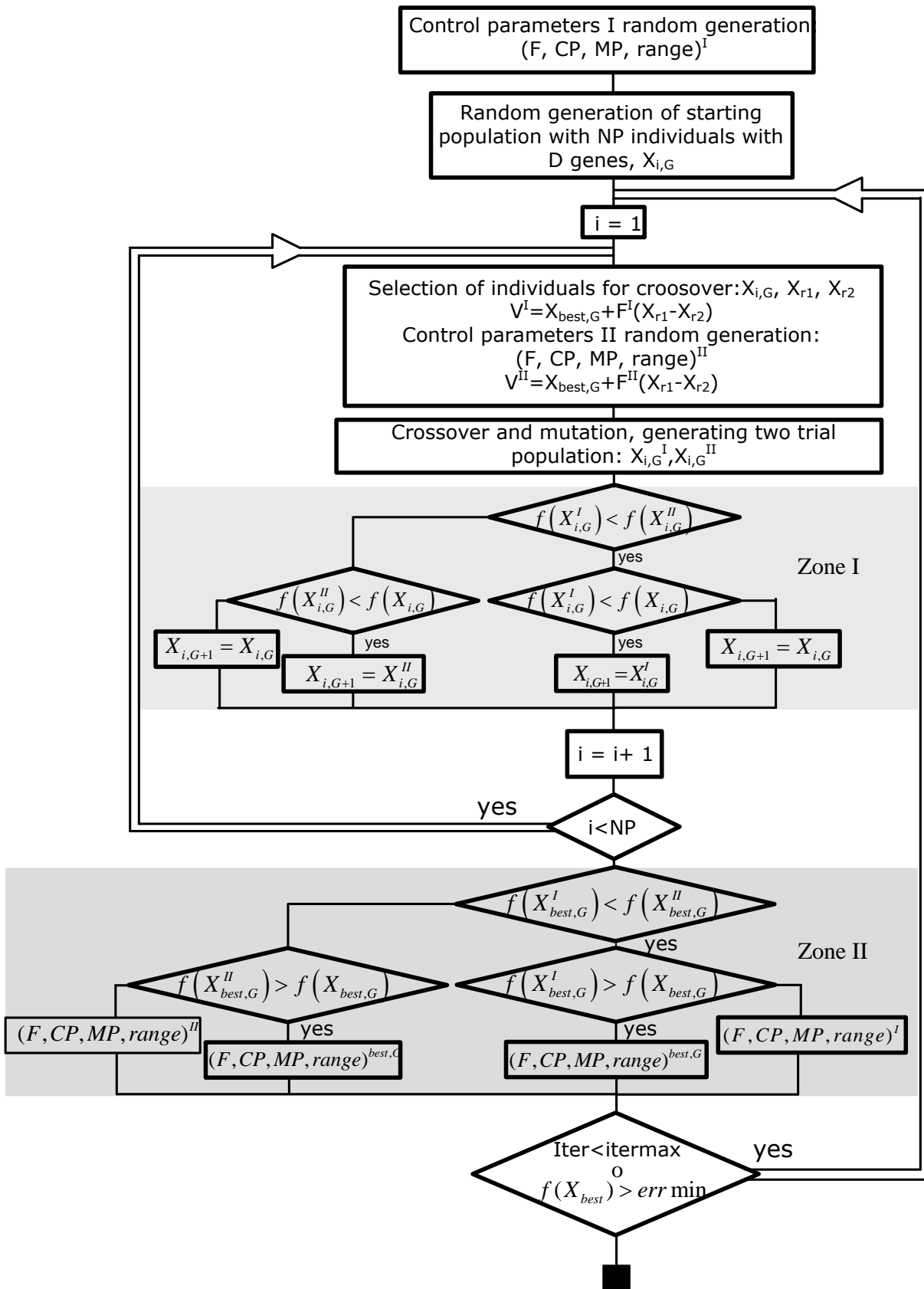
**Figure 1: discrete multipoint crossover for reproduction.**







**Figure 4: reproduction followed by mutation.**

Figure 5: general Scheme of IOA<sup>s-at</sup>

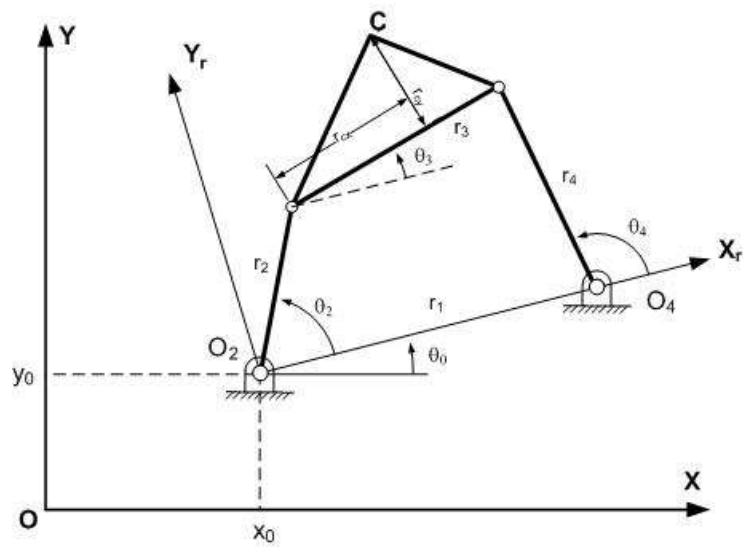


Figure 6: design variables of four-bar mechanism for the first objective function.

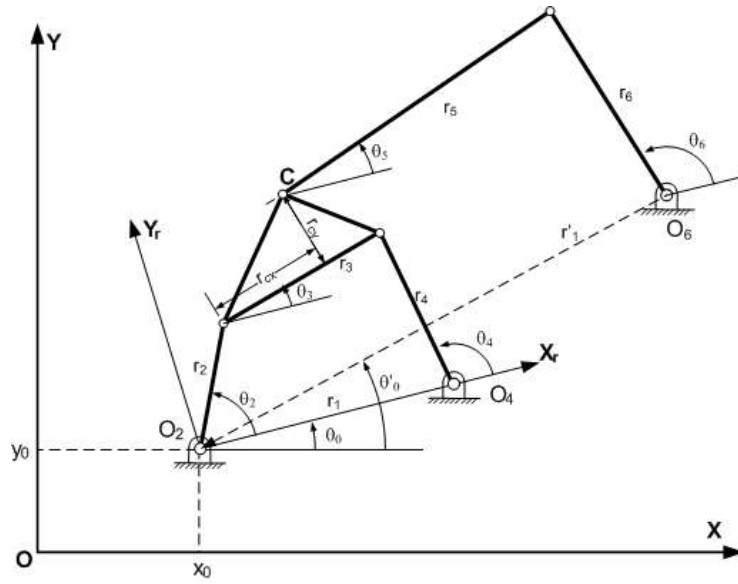


Figure 7: design variables of a six-bar mechanism for the second objective function.

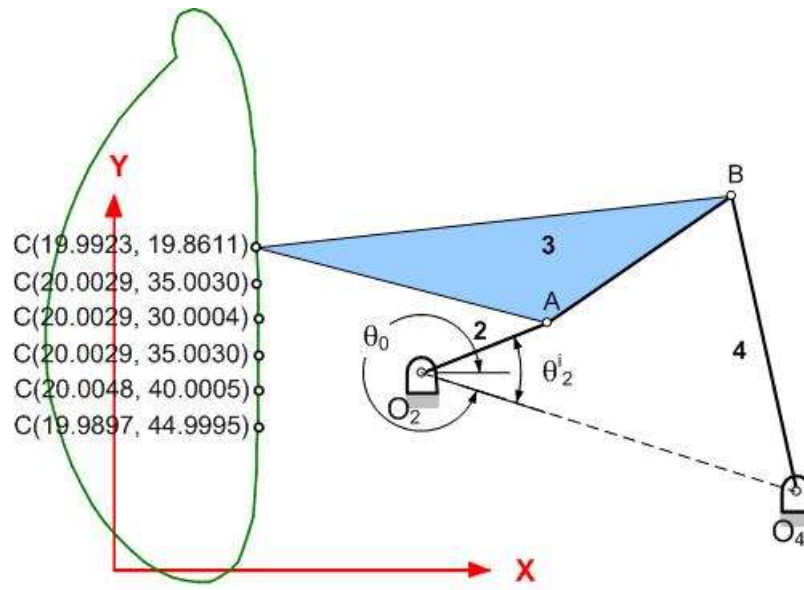


Figure 8: best mechanism for Case 1.

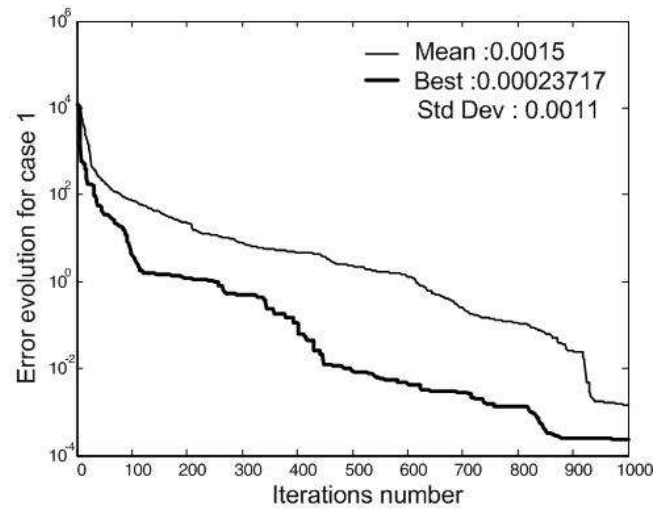


Figure 9: evolution of the best and the mean for 50 runs for Case 1.

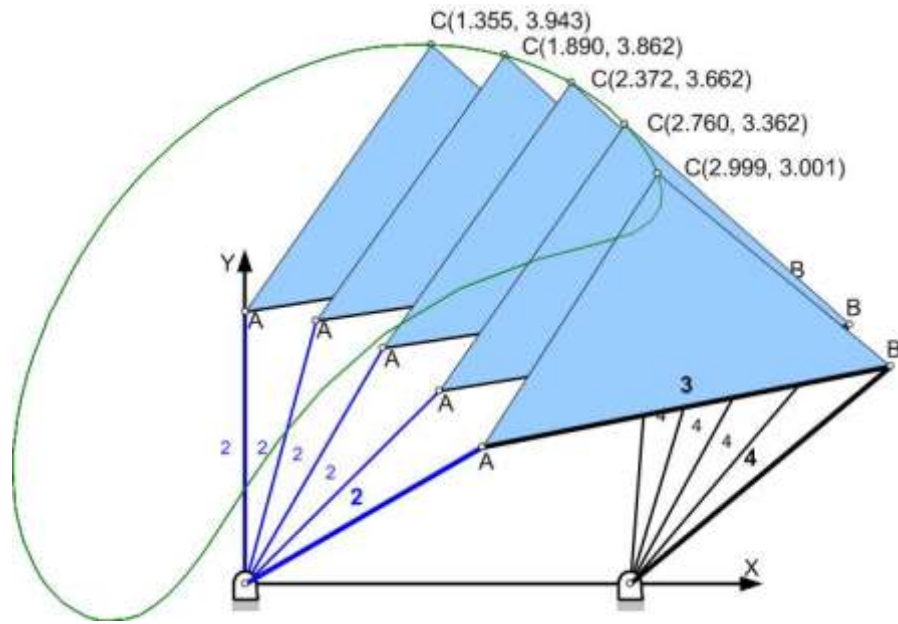


Figure 10: best Mechanism for Case 2.

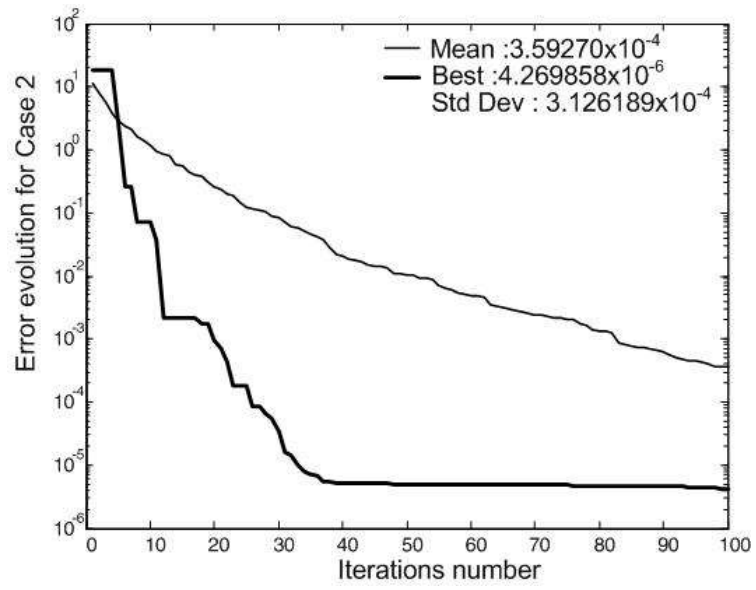


Figure 11: error evolution for Case 2, for the best and for the mean.

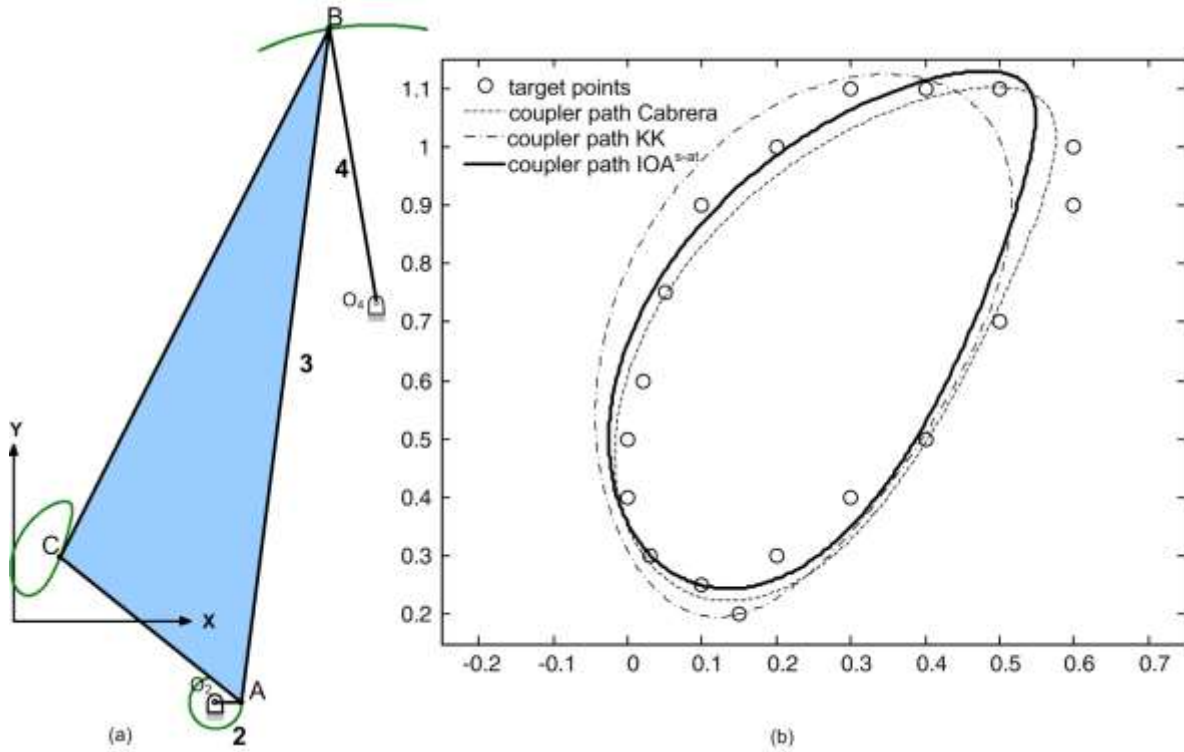


Figure 12: (a): best mechanism found for Case 3 and (b): coupler paths.

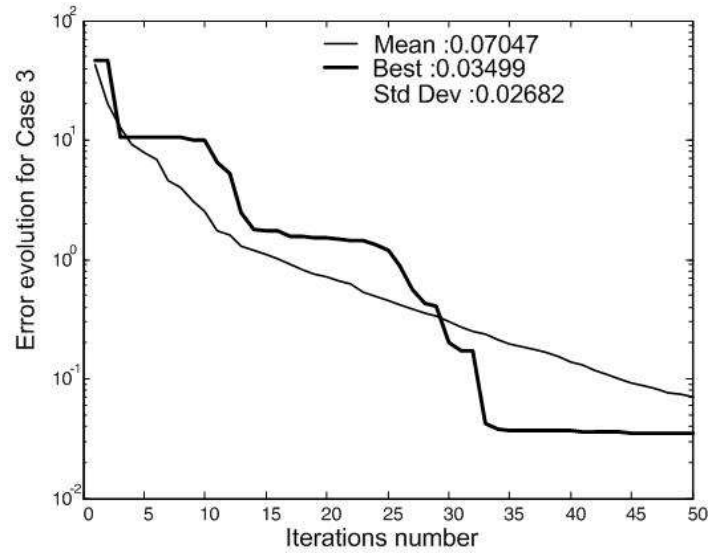
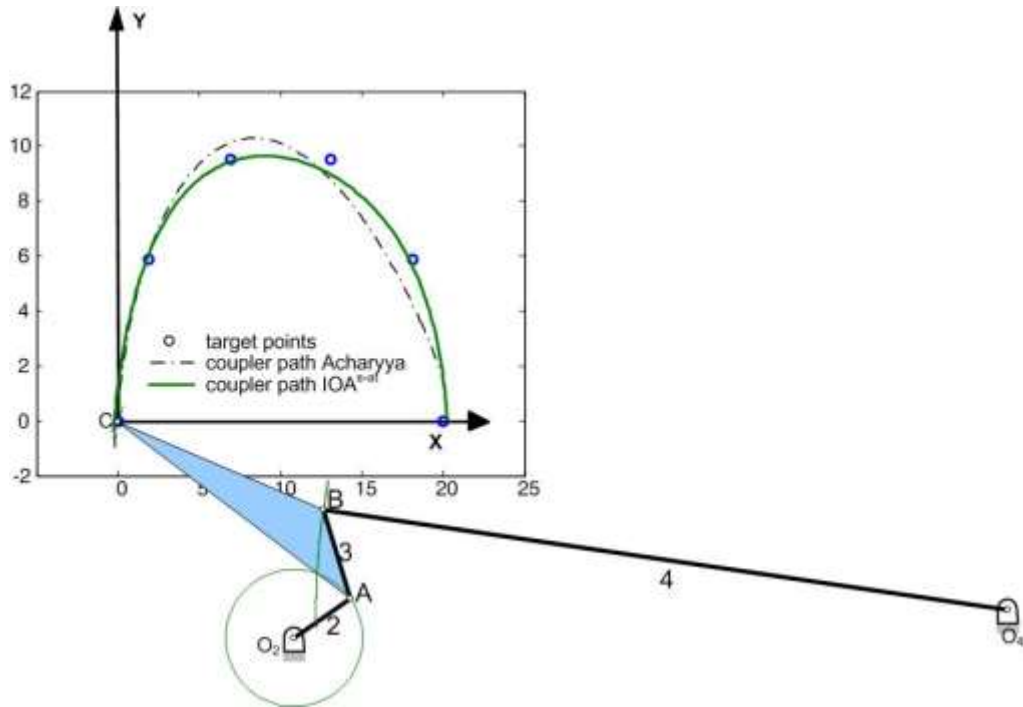
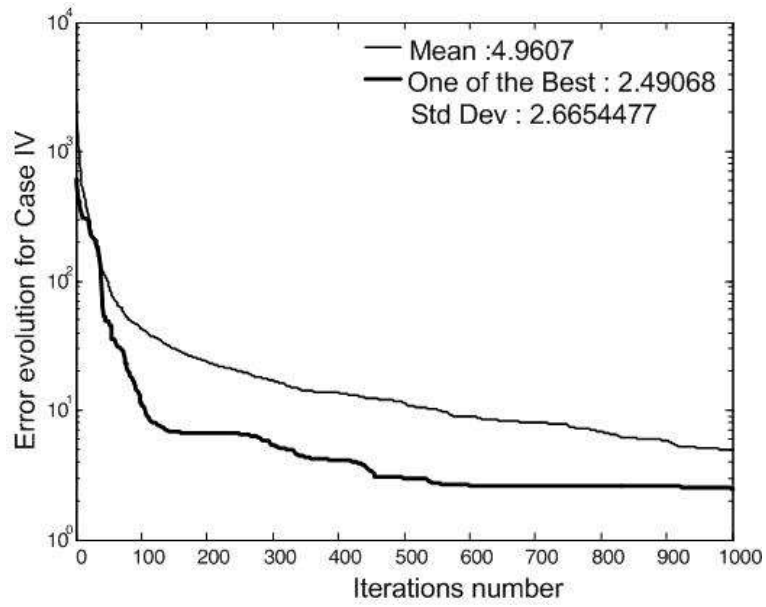


Figure 13: error evolution for Case 3, for the best and for the mean.





**Figure 15: error evolution for Case 4.**

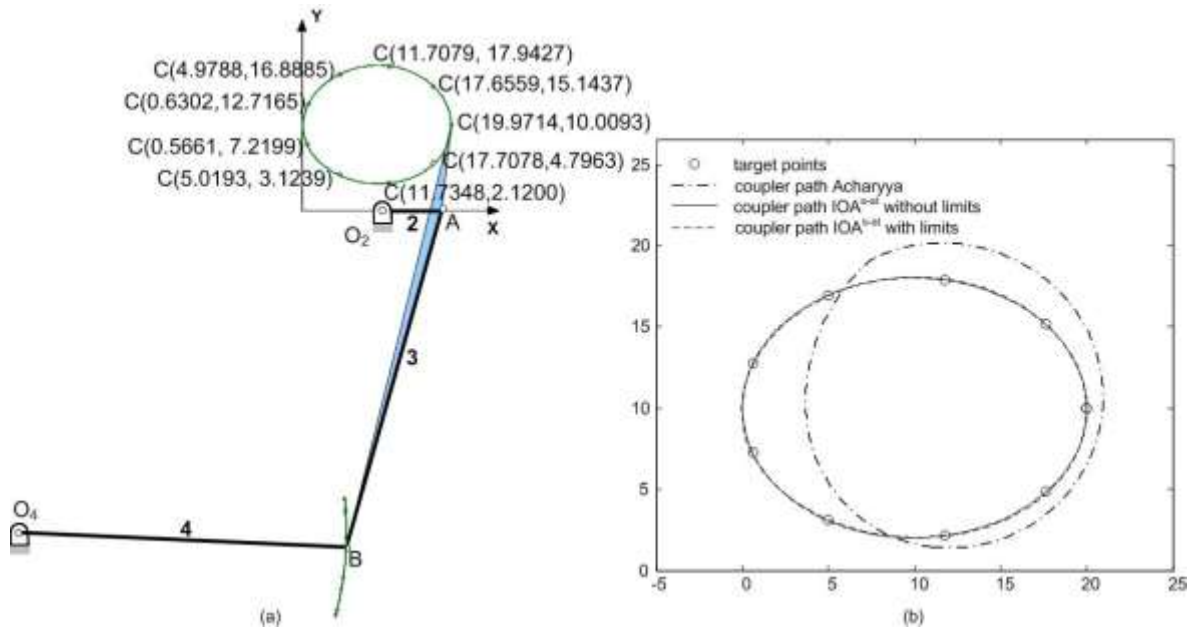


Figure 16: (a) best mechanism found with limits of variables, (b) elliptical path trajectory for every algorithm.

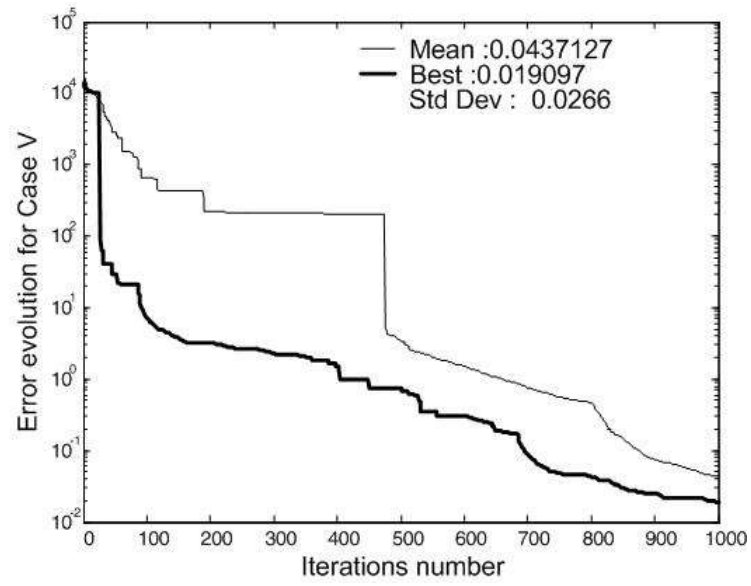


Figure 17: IOA<sup>s-at</sup> error evolution of the best and the mean for Case 5.

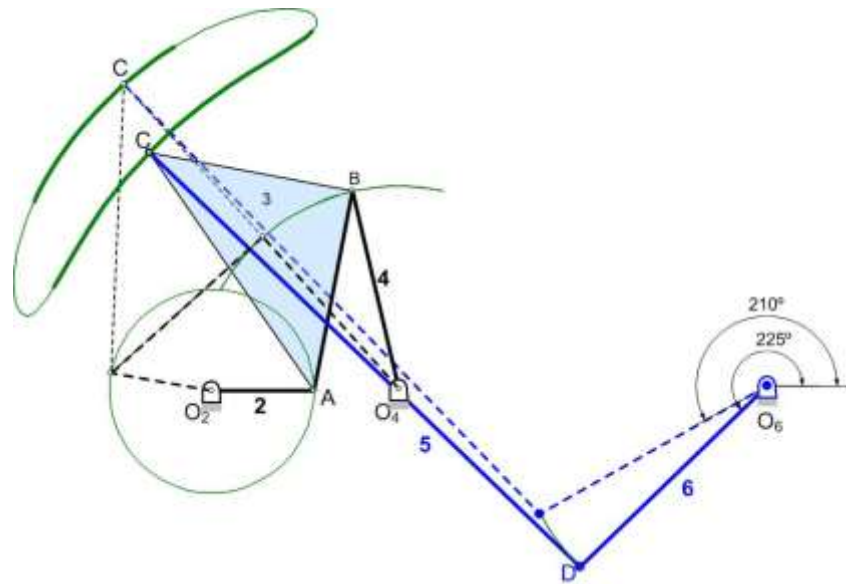


Figure 18: coupler trajectory and dwells of the six-bar found using IOA<sup>s-at</sup>.

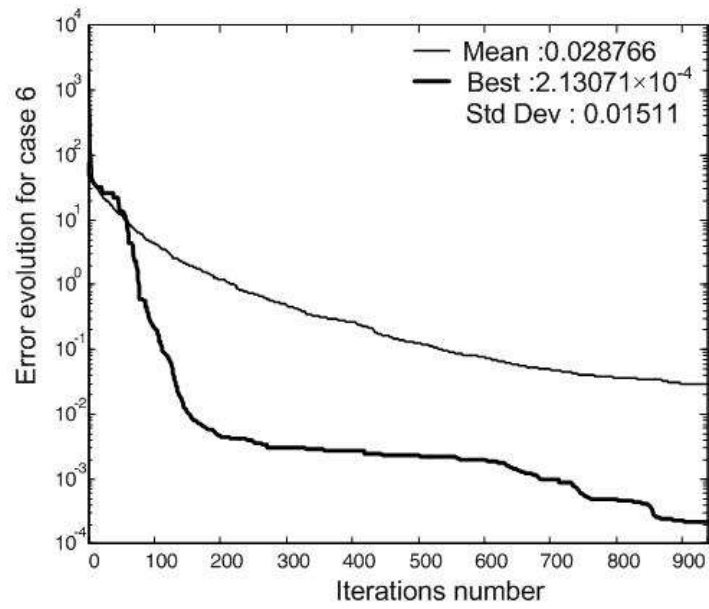
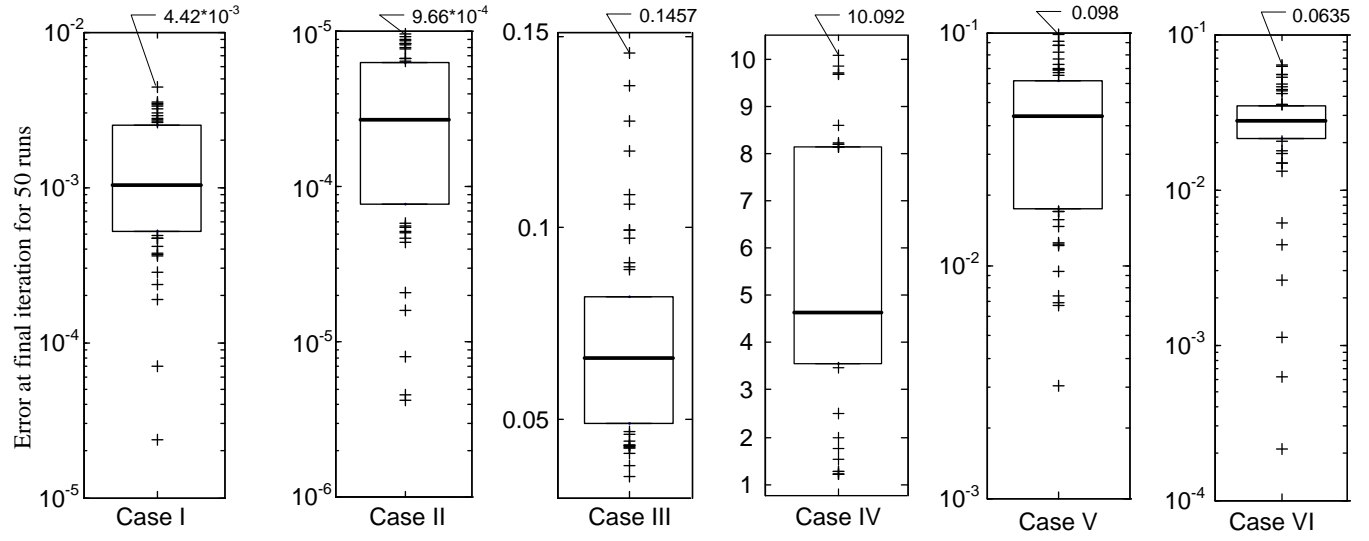
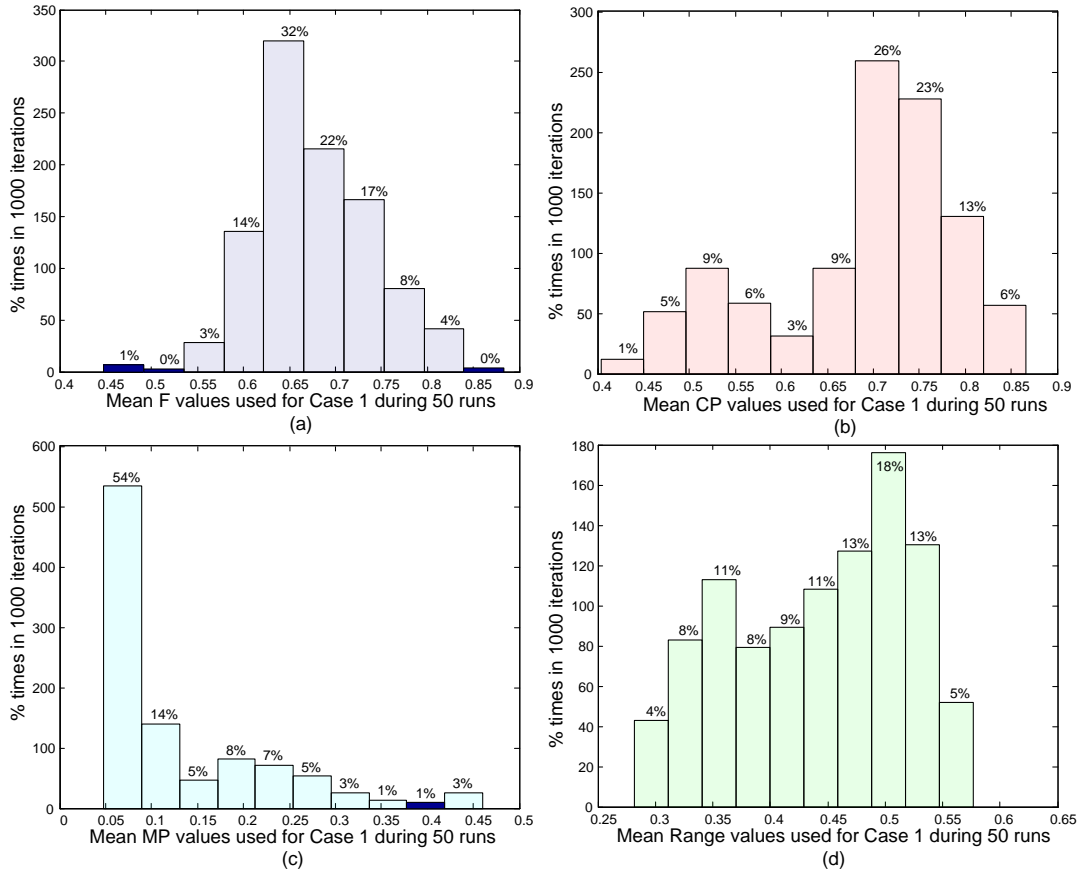


Figure 19: IOA<sup>s-at</sup> error evolution of the best and the mean for Case 6.



**Figure 20: statistical study for the six cases studied.**



**Figure 23: values and times rate that this value occurs of F, CP, MP, Range for Case 1.**